

UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA: INGENIERÍA DE SISTEMAS

Tesis previa a la obtención del título de: INGENIERO DE SISTEMAS

TEMA:

**DISEÑO Y CONSTRUCCIÓN DE UN ROBOT SCARA QUE JUEGA
DAMAS POR MEDIO DE VISIÓN ARTIFICIAL PROGRAMADO EN JAVA.**

AUTOR:

EDGAR HERNÁN CHANGOLUISA LASLUIA

DIRECTOR:

GONZALO BAYARDO CAMPUZANO NIETO

Quito, junio de 2014

**DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO
DEL TRABAJO DE TITULACIÓN**

Yo, autorizo a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además declaro que los conceptos, análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad del autor.

EDGAR HERNÁN CHANGOLUISA LASLUIA

CI: 1717164709

DEDICATORIA

Al redactar estas líneas me invade una mezcla de emociones, porque este trabajo es la culminación de una meta que he anhelado y esperado.

El trabajo lo dedico con todo mi amor y cariño a mi Dios, a mi mami Melida, a mis hermanas Consuelo y María Belén, a mis sobrinos Joselyn, Isaac, David, Emily, a mi padre Edgar. Gracias por su amor, paciencia, apoyo, experiencias de vida y valiosos consejos, ustedes son una bendición en mi vida.

AGRADECIMIENTO

Al Fis. Gonzalo Bayardo Campuzano Nieto, mi director de trabajo de titulación, gracias de corazón por su confianza, apoyo, motivación y paciencia.

A la lectora Mgs. Lina Patricia Zapata Molina por sus valiosas correcciones que me ayudaron a mejorar este trabajo y por su amistad.

Para todos mis profesores de la Universidad Politécnica Salesiana, gracias por sus valiosos conocimientos.

ÍNDICE

| | |
|--|----------|
| INTRODUCCIÓN | 1 |
| CAPÍTULO 1..... | 2 |
| ASPECTOS GENERALES..... | 2 |
| 1.1. Planteamiento del problema | 2 |
| 1.2. Objetivos | 3 |
| 1.2.1. Objetivo general..... | 3 |
| 1.2.2. Objetivos específicos | 3 |
| 1.3. Justificación del proyecto | 4 |
| 1.4. Alcance del proyecto | 5 |
| 1.5. Metodologías | 6 |
| 1.6. Procedimiento de investigación..... | 7 |
| 1.7. Pasos para el procedimiento de investigación | 7 |
| 1.8. Técnicas de recolección..... | 7 |
| CAPÍTULO 2..... | 9 |
| VISIÓN ARTIFICIAL | 9 |
| 2.1. Generalidades | 9 |
| 2.2. Elementos básicos de un sistema de visión artificial..... | 10 |
| 2.3. Iluminación..... | 11 |
| 2.3.1. Tipos de reflexión. | 11 |
| 2.3.1.1. Reflexión especular..... | 12 |
| 2.3.1.2. Reflexión difusa. | 12 |
| 2.3.2. Técnicas de iluminación. | 13 |
| 2.3.2.1. Iluminación posterior (backlight)..... | 13 |

| | | |
|--|---|-----------|
| 2.3.2.2. | Iluminación campo oscuro. | 14 |
| 2.3.2.3. | Iluminación frontal direccional. | 15 |
| 2.3.2.4. | Iluminación frontal. | 15 |
| 2.3.3. | Fuentes de luz. | 16 |
| 2.4. | Dispositivos de captura y de adquisición de datos | 17 |
| 2.4.1. | Webcam. | 18 |
| 2.4.1.1. | Cámara digital. | 18 |
| 2.4.1.2. | Software. | 19 |
| 2.5. | Algoritmos de análisis de imágenes y reconocimientos de formas | 20 |
| 2.5.1. | Filtrado o pre-procesamiento. | 20 |
| 2.5.1.1. | Conversión de una imagen a color a escala de grises. | 20 |
| 2.5.1.2. | Operaciones aritméticas de imágenes – Diferencia. | 21 |
| 2.5.1.3. | Filtro de suavizado: mediana. | 22 |
| 2.5.1.4. | Filtro de suavizado: Gaussiano. | 23 |
| 2.5.2. | Segmentación o aislamiento de los objetos de interés. | 24 |
| 2.5.2.1. | Algoritmo de Canny. | 24 |
| 2.5.2.2. | Dilatación. | 26 |
| 2.5.2.3. | Etiquetado. | 27 |
| CAPÍTULO 3..... | | 29 |
| INTELIGENCIA ARTIFICIAL , ALGORITMOS PARA JUEGOS..... | | 29 |
| 3.1. | Introducción..... | 29 |
| 3.2. | ¿Qué es la inteligencia? | 30 |
| 3.3. | ¿Qué es la inteligencia artificial? | 31 |
| 3.4. | Juegos | 31 |

| | | |
|--|---|-----------|
| 3.4.1. | Métodos de análisis de juegos por computadora. | 33 |
| 3.4.1.1. | Búsqueda minimax. | 33 |
| 3.4.1.2. | Poda alfa-beta de la búsqueda minimax. | 35 |
| 3.4.1.3. | Toma de decisiones en tiempo real. | 37 |
| 3.4.1.3.1. | Funciones de evaluación. | 38 |
| 3.4.1.3.2. | Corte de la búsqueda. | 39 |
| CAPÍTULO 4..... | | 41 |
| ROBÓTICA- ROBOT SCARA | | 41 |
| 4.1. | Introducción..... | 41 |
| 4.2. | Definición..... | 42 |
| 4.3. | Robot scara | 43 |
| 4.3.1. | Estructura mecánica..... | 44 |
| 4.3.1.1. | Articulaciones. | 45 |
| 4.3.1.2. | Eslabón..... | 46 |
| 4.3.1.3. | Grados de libertad. | 46 |
| 4.3.2. | Dimensiones del robot scara. | 47 |
| 4.3.3. | Espacio de tarea. | 48 |
| 4.3.4. | Actuadores. | 49 |
| 4.3.5. | Actuadores electromagnéticos | 49 |
| 4.3.5.1. | Servomotores. | 49 |
| 4.3.6. | Cinemática inversa..... | 52 |
| CAPÍTULO 5..... | | 54 |
| ANÁLISIS Y DISEÑO DEL SISTEMA | | 54 |
| 5.1. | Definición de requisitos..... | 54 |

| | | |
|--------------------------------------|--|-----------|
| 5.1.1. | Requerimientos funcionales..... | 54 |
| 5.1.2. | Requisitos no funcionales. | 55 |
| 5.2. | Diseño del sistema..... | 55 |
| 5.2.1. | Diseño de hardware. | 56 |
| 5.2.1.1. | Diseño mecánico. | 56 |
| 5.2.1.2. | Sistema electrónico. | 57 |
| 5.2.1.3. | Sistema de control. | 58 |
| 5.2.2. | Diseño software. | 59 |
| 5.2.2.1. | Diagrama de casos de uso. | 59 |
| 5.2.2.2. | Diagrama de secuencia..... | 62 |
| 5.2.2.3. | Diagrama de clases..... | 65 |
| 5.2.2.4. | Diagrama de actividades. | 74 |
| 5.2.2.5. | Diagrama de despliegue. | 76 |
| CAPÍTULO 6..... | | 77 |
| CONSTRUCCIÓN DEL SISTEMA..... | | 77 |
| 6.1. | Entorno de trabajo | 77 |
| 6.2 | Implementación de los módulos..... | 79 |
| 6.2.1 | Módulo de robótica..... | 80 |
| 6.2.1.1. | Diseño mecánico. | 80 |
| 6.2.1.2. | Sistema electrónico | 81 |
| 6.2.1.3. | Servomotores. | 84 |
| 6.2.2. | Programación del software..... | 86 |
| 6.2.2.1. | Definición del algoritmo de búsqueda. | 86 |
| 6.2.3. | Procesamiento y análisis de imágenes..... | 88 |

| | |
|--|------------|
| 6.2.4. Aplicación de filtros. | 89 |
| 6.2.5. Detección de bordes. | 91 |
| 6.2.6. Dilatación de bordes. | 91 |
| 6.2.7. Reconocimiento de la jugada realizada. | 92 |
| 6.2.8. Interfaz de usuario del sistema. | 94 |
| 6.3. Pruebas y ajustes | 94 |
| 6.4. Resultados obtenidos | 100 |
| CONCLUSIONES..... | 101 |
| LISTA DE REFERENCIAS | 102 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1. Elementos básicos que componen un sistema de visión artificial | 10 |
| Figura 2. Reflexión especular | 13 |
| Figura 3. Reflexión difusa..... | 13 |
| Figura 4. Iluminación posterior..... | 14 |
| Figura 5. Iluminación campo oscuro..... | 14 |
| Figura 6. Iluminación frontal direccional..... | 15 |
| Figura 7. Iluminación frontal | 16 |
| Figura 8. Funcionamiento interno de una cámara digital..... | 18 |
| Figura 9. Esquema de un CCD que utiliza un filtro Bayer | 19 |
| Figura 10. Pasos de un proceso de análisis y procesamiento de Imágenes..... | 20 |
| Figura 11. Componentes RGB de una imagen..... | 21 |
| Figura 12. Resta de dos imágenes | 22 |
| Figura 13. Aplicación del filtro de la mediana..... | 22 |
| Figura 14. Aplicación del filtro Gaussiano | 23 |
| Figura 15. Ejemplo de dilatación de bordes de una imagen. | 27 |
| Figura 16. Procesamiento de etiquetado | 28 |
| Figura 17. Árbol de juegos minimax..... | 34 |
| Figura 18. Algoritmo minimax | 35 |
| Figura 19. Aplicación de la poda alfa-beta | 36 |
| Figura 20. El algoritmo de la búsqueda alfa-beta | 37 |
| Figura 21. Búsqueda alfa y beta con límite de profundidad fija. | 39 |
| Figura 22. Arquitectura de un robot industrial..... | 43 |
| Figura 23. Robot scara 3 grados de libertad..... | 45 |

| | |
|---|----|
| Figura 24. Simbología de la articulación de rotación | 46 |
| Figura 25. Simbología articulación prismática | 46 |
| Figura 26. Ejemplo de grados de libertad | 47 |
| Figura 27. Dimensiones y parámetros del robot scara | 47 |
| Figura 28. Área de trabajo robot scara | 48 |
| Figura 29. Sistema básico de control de posición | 51 |
| Figura 30. Pulsos para control de un servomotor..... | 51 |
| Figura 31. Problema de la cinemática inversa de un robot scara | 52 |
| Figura 32. Ejes y campos de rotación del robot scara..... | 56 |
| Figura 33. Módulos propuesto para el control | 57 |
| Figura 34. Diagrama de flujo del modelo de cinemática inversa..... | 59 |
| Figura 35. Casos de uso | 60 |
| Figura 36. Diagrama de secuencia | 62 |
| Figura 37. Diagrama de clases | 65 |
| Figura 38. Diagrama de actividades..... | 75 |
| Figura 39. Diagrama de despliegue..... | 76 |
| Figura 40. Prototipo y sus componentes | 77 |
| Figura 41. Prototipo robot scara..... | 80 |
| Figura 42. Diagrama esquemático del circuito de la Fuente de alimentación | 82 |
| Figura 43. Diagrama esquemático del sistema de control..... | 82 |
| Figura 44. Diagrama PCB de la tarjeta madre del sistema | 83 |
| Figura 45. Circuito de construcción de la tarjeta madre (vista frontal) | 83 |
| Figura 46. Cálculo del torque para un robot | 85 |
| Figura 47. Proceso de extraer los objetos del fondo | 88 |

| | |
|--|----|
| Figura 48. Imagen luego de aplicar filtro de la mediana 15x15..... | 89 |
| Figura 49. Imagen luego de aplicar filtro gauss 7 | 90 |
| Figura 50. Obtención de bordes de la imagen..... | 90 |
| Figura 51. Imagen con los bordes dilatados..... | 92 |
| Figura 52. Reconocimiento de una jugada realizada por el participante humano | 92 |
| Figura 53. Pruebas de precisión de posicionamiento den el plano xy | 94 |
| Figura 54. Pruebas del filtro de mediana | 97 |
| Figura 55. Filtro de suavizado de Gauss | 98 |
| Figura 56. Pruebas del algoritmo de canny para buscar bordes..... | 99 |
| Figura 57. Pruebas de dilatación de bordes de las fichas | 99 |

ÍNDICE TABLAS

| | |
|---|----|
| Tabla 1. Definición de inteligencia artificial , organizada en 4 categorías. | 30 |
| Tabla 2. Dimensiones robot scara | 48 |
| Tabla 3. Características técnicas | 56 |
| Tabla 4. Actores del sistema | 60 |
| Tabla 5. Especificación del caso de uso: Iniciar juego | 61 |
| Tabla 6. Especificación del caso de uso: Mover ficha | 61 |
| Tabla 7. Descripción de los pasos del diagrama de secuencia | 63 |
| Tabla 8. Detalle del diagrama de clase, clase Image..... | 66 |
| Tabla 9. Detalle del diagrama de clase, clase Imagen Controlador | 67 |
| Tabla 10. Detalle del diagrama de clase, clase Juego Damas | 68 |
| Tabla 11. Detalle del diagrama de clase, clase Jugador | 69 |
| Tabla 12. Detalle del diagrama de clase, clase Ficha..... | 70 |
| Tabla 13. Detalle del diagrama de clase, clase TableroJuego..... | 71 |
| Tabla 14. Detalle del diagrama de clase, clase Humano..... | 72 |
| Tabla 15. Detalle del diagrama de clase, clase Robot..... | 73 |
| Tabla 16. Detalle del diagrama de clase, clase Robot Controlador | 74 |
| Tabla 17. Dimensiones de los elementos del prototipo del robot scara | 80 |
| Tabla 18. Características de los servomotores usados en el prototipo | 84 |
| Tabla 19. Pruebas de presión de posición del robot scara en el eje xy | 95 |
| Tabla 20. Prueba del nivel óptimo del juego de damas..... | 96 |

ÍNDICE DE ECUACIONES

| | |
|--|----|
| Ecuación 1. Expresión matemática de la iluminación | 21 |
| Ecuación 2. Operaciones aritméticas de imágenes - diferencia | 21 |
| Ecuación 3. Función gausseana..... | 23 |
| Ecuación 4. Ejemplo de un filtro gaussiano 5x5 y $\sigma = 1.4$ | 25 |
| Ecuación 5. Ecuaciones para encontrar gradiente y dirección de la imagen | 25 |
| Ecuación 6. Ecuación de dilatación (1)..... | 26 |
| Ecuación 7. Ecuación de dilatación (2)..... | 26 |
| Ecuación 8. Función lineal de Shanon | 33 |
| Ecuación 9. Contribuciones numéricas de cada característica..... | 39 |
| Ecuación 10. Ángulos de rotación de eslabones | 44 |
| Ecuación 11. Teorema del coseno..... | 53 |
| Ecuación 12. Formulas de la resolución geométrica de la cinemática inversa | 53 |
| Ecuación 13. Cálculo de la precisión de posicionamiento | 95 |

ÍNDICE DE ANEXOS

| | |
|--|-----|
| Anexo 1: Reglas de las damas..... | 106 |
| Anexo 2: Lista de materiales para el circuito de la tarjeta madre | 108 |
| Anexo 3: Costo del diseño y construcción del prototipo del brazo robótico scara. . | 108 |

RESUMEN

El trabajo de investigación y desarrollo tiene como propósito construir un brazo robótico industrial, tipo scara, que pueda jugar damas contra un oponente humano de la manera más natural posible. Para lo cual se combinó tres áreas tecnológicas como son:

- Visión por computadora para la captura de los movimientos que el contrincante realice sobre el tablero de damas.
- Se usará la inteligencia artificial para la implementación de un algoritmo de búsqueda, para determinar la mejor jugada.
- La construcción del brazo robótico se determinara dependiendo de los movimientos de las fichas dentro del tablero de damas.

ABSTRACT

The research and development aims to build an industrial robotic arm scara type, you can play checkers against a human opponent in the most natural way possible. For which three technology areas are combined as:

- Computer vision to capture the opponent's movements made on the checkerboard.
- Artificial intelligence for the implementation of a search algorithm is used, to determine the best move.
- The construction of the robotic arm is determined depending on the movements of the tabs within the checkerboard.

INTRODUCCIÓN

En el proyecto, se va a presentar una aplicación del juego de las damas, donde uno de los jugadores es un robot. El robot no solamente piensa y calcula las jugadas, sino las culmina realizando el desplazamiento de las fichas como si fuese un jugador humano (IEEE, 2008, pág. 42).

El primer problema al que se enfrenta es a la integración de las diferentes tecnologías como son Robótica, inteligencia artificial y visión artificial. Este problema es cada vez más creciente y la robótica es un ejemplo viviente. En este sentido, nuestra aplicación muestra un grado de integración más que notable como veremos más adelante (IEEE, 2008, pág. 42).

En la literatura no abundan aplicaciones donde un robot participa en un juego. Sin embargo, puede señalar que el primer robot que juega a las damas fue desarrollado en la universidad de Rochester Oklahoma en 1991 (IEEE, 2008, pág. 43). El robot, conocido por el nombre de LEFTY, se encuentra hoy en el museo de “the Omniplex Science Museum”. El robot está dotado de un sistema de visión y de un sistema de voz. El objetivo de LEFTY era mostrar una aplicación con una integración de diferentes tecnologías como la robótica, la visión artificial, la inteligencia artificial. (IEEE, 2008, pág. 42).

El sistema básicamente consta de 3 módulos que interactúan entre si estos son: visión por computadora, inteligencia artificial y robótica. Cada módulo antes mencionado cumple funciones específicas y actúan en un ambiente controlado.

Para que un ser humano pueda interactuar con el sistema se construyó un prototipo de un brazo robótico tipo scara que pueda jugar una partida de damas por medio de visión artificial de la manera más natural posible.

CAPÍTULO 1

ASPECTOS GENERALES

1.1. Planteamiento del problema

Desde el comienzo de la década de los 40 con la evolución de la informática, constantemente se han hecho investigaciones científicas y tecnológicas con la finalidad de facilitar algunas actividades propias de los seres humanos. Se ha logrado automatizar muchos procesos mecánicos, de cálculo, de almacenamiento de datos, de procesamiento, etc. Desarrollando, cada vez, herramientas de cómputo capaces de auxiliar en forma directa cada una de estas actividades.

En resumen se puede decir que la inteligencia artificial desarrolla maneras de emular el procesamiento de la información humana a través de las computadoras o robots. Los avances tecnológicos han logrado obtener procesadores de gran capacidad que les permite “pensar” y “aprender”.

Gran parte de la investigación de la inteligencia artificial ha estado centrada a la creación de algoritmos de razonamiento y aprendizaje que permitan a los robots realizar tareas repetitivas que realizan los seres humanos, una de las más comunes es que puedan resolver juegos matemáticos e incluso juegos de mesa como son las Damas.

Las damas, es un juego sencillo tal así que persona lo puede aprender a jugar, pero tiene su complejidad debido a las reglas de las versiones y en qué país se juega, adicionalmente en un tablero 8x8 con 24 ficha las posibles jugadas que se pueden realizar son 10 elevado a la 20 aproximadamente.

Otra rama de la inteligencia artificial es la visión por computadora la misma que tiene la finalidad de modelar matemáticamente una imagen para que pueda ser analizada e interpretada. El ser humano aprende a usar el sentido de la vista desde cuando es bebe y es uno de los sentidos más importantes, pues le sirve para reconocer el entorno en el que se encuentra, este sentido es el más utilizado y del que menos se conoce como se produce debido a su gran complejidad.

Para imitar las habilidades que tienen los seres humano de manipular los objetos que se sitúan a su alrededor así como también movilizarse en el medio ambiente que se rodea, se ha creado el robot móvil, el cual es un dispositivo programable que tiene determinados grados de movilidad para realizar tareas y adaptarse al ambiente en el que opera.

Con las aclaraciones anteriores el trabajo investigativo pretende resolver las siguientes inquietudes:

- ¿Cuál es el algoritmo ideal para elegir los mejores movimientos y poder ganar el juego de damas?
- ¿Cómo se puede reconocer, visualmente desde la PC, cual fue la jugada que el contrincante realizo?
- ¿Se puede construir un prototipo de robot scara, que sea lo suficientemente preciso, rápido y de bajo costo, que pueda colocar las fichas correctamente en el tablero?

1.2. Objetivos

1.2.1. Objetivo general.

Diseñar un brazo robótico scara que pueda jugar una partida de damas contra un ser humano, por medio de visión artificial, dicho brazo deberá conocer la jugada de su contrincante, utilizando el lenguaje de programación Java (Jdk 1.7).

1.2.2. Objetivos específicos.

- Construir e implementar un módulo de reconocimiento de imágenes en movimiento utilizando la librería de visión artificial de Intel denominada OpenCV, que nos va a permitir reconocer las jugadas realizadas por el contrincante en el tablero de damas.
- Diseñar y construir un brazo robótico tipo scara de bajo costo que pueda movilizar las fichas de una manera rápida y precisa para que el contrincante no sienta un tedio cuando juegue la partida de damas.

- Elaborar un algoritmo de búsqueda, evaluación y de toma de decisiones enfocado en el juego de damas que calcule si el próximo movimiento a ejecutarse va hacer el ganador, perdedor o puede terminar en tablas, con la finalidad de ganar el Juego.
- Diseñar una interfaz gráfica amigable con el usuario que integre todos los módulos que intervienen en el proceso y adicionalmente permita la configuración de los mismos.

1.3. Justificación del proyecto

El proyecto tiene la finalidad de aplicar los conocimientos de robótica, visión e inteligencia artificial, adquiridos durante el transcurso de la carrera de Ingeniería de Sistemas, para lo cual se va a realizar una demostración del juego de damas donde uno de los jugadores es un robot, quien no solamente "piensa" y "analiza" las jugadas, sino que puede “ver” las jugadas del contrincante y realizar el desplazamiento de las fichas de manera muy similar un jugador humano.

El primer reto a enfrentar, es el tratar de integrar los módulos de robótica: visión e inteligencia artificial utilizando Java, por lo general se utiliza para el desarrollo de aplicaciones de ambiente web, no existe documentación oficial por parte del propietario de este lenguaje de programación¹ acerca de las librerías que permitan desarrollar tecnología para los módulos antes mencionados, la mayor parte de la documentación encontrada es de instituciones dedicadas a la investigación que han utilizado este lenguaje.

La literatura no es abundante acerca de aplicaciones donde un robot participa en un juego, sin embargo, se señala que el primer robot que jugo a las damas fue desarrollado en la universidad de Rochester, Oklahoma en 1991. El robot conocido por el nombre de Lefty (Aliane, 2004, pág. 1), se encuentra hoy en el museo de the Omniplex Science Museum. El robot está dotado de un sistema de visión y de un sistema de voz. El objetivo de LEFTY era mostrar una aplicación con una

¹ Java fue creado por Sun Microsystem, que fue adquirida por grupo de software Oracle.

integración de diferentes tecnologías como la robótica, la visión artificial, la inteligencia artificial (Aliane, 2004, pág. 1).

Para cumplir el objetivo del proyecto se va a diseñar y construir un robot scara para mover las fichas dentro del tablero de ajedrez, un sistema de visión artificial que consta de una cámara web colocada sobre el tablero y que tiene un sistema de iluminación para que la luz del exterior no influya en el análisis de las imágenes y un ordenador personal el cual va a ejecutar la interfaz para controlar todo la aplicación.

1.4. Alcance del proyecto

El proyecto va hacer un prototipo, el mismo que no va hacer donado a la universidad; el estudio se limitará a la creación física de un robot a scara, con tres grados de libertad, no va hacer un manipulador, para poder mover la ficha va utilizar un electroimán y siempre va a mantener una posición de inicio a la cual va a regresar luego de realizar cada jugada.

El sistema operativo donde va a ejecutarse la aplicación es Windows XP de 32 bits, debido a que Java no tiene librerías nativas que permita el manejo y configuración de los puertos USB en 64 bits.

El lenguaje de programación donde se va a desarrollar la interfaz de usuario que va a permitir la ejecución y configuración del sistema es Java utilizando como IDE Netbeans 7.4 y como compilador JDK 1.7.

La aplicación va a tener un algoritmo para jugar a las damas de acuerdo a las reglas básicas (Anexo 1), este algoritmo tiene la finalidad buscar y elegir los movimientos que permitan ganar el juego, esto no quiere decir que siempre lo pueda hacer, los factores a tomar en cuenta son la memoria y procesador de la computadora donde se ejecute el programa.

Para la visión artificial se va a utilizar una cámara web que ayudará con la obtención de las imágenes, se va a proveer de una buena iluminación que va a estar situada sobre el tablero y de manera fija, lo que ayudara a mejorar el procesamiento y análisis de las imágenes.

El reconocimiento de imágenes no va a hacer en tiempo real, va a existir un retardo de unos cuantos milisegundos hasta que el sistema pueda analizar que jugada realizó el contrincante, este evento va a ser ejecutado luego de que el contrincante haya realizado un movimiento y lo informe por medio de un botón que va a estar colocado en la interface de usuario.

1.5. Metodologías

La metodología indica el camino a seguir en una investigación, recuerde que metodología aparece de la palabra método que significa camino o guía, por lo cual es importante definir cuál será la metodología a seguir en el proyecto de tal forma que se optimice recursos y minimice el tiempo de implementación.

Los métodos a utilizarse para el desarrollo del plan del trabajo será el método inductivo - deductivo, y la investigación de campo.

- La inducción es un proceso mediante el cual se parte de conocimientos particulares a uno general, esto va a hacer de gran ayuda en la formulación de la hipótesis, este proceso de investigación está siempre vinculado a la deducción.
- La deducción es un proceso que se basa en las afirmaciones y generalizaciones a partir del cual se realizan pruebas o deducciones individuales, es decir que se pasa de un conocimiento general a uno más específico.

En la actividad científica la inducción y la deducción se complementan entre sí del estudio de numerosos casos particulares, a través de la inducción se llega a determinar generalizaciones, leyes empíricas, las que constituyen puntos de partida para definir o confirmar formulaciones teóricas, de dichas formulaciones teóricas se deducen nuevas conclusiones lógicas, las que son sometidas a comprobaciones experimentales. Solamente la complementación mutua entre estos procedimientos puede proporcionar un conocimiento verdadero sobre la realidad (Hernández, 2006, pág. 36).

- La investigación de campo es un estudio que se realiza en el medio donde se desarrolla el problema que se está tratando de resolver.

1.6. Procedimiento de investigación

Si bien el qué?, por qué? y el para qué? pueden ser delimitados en un proyecto de investigación, el cómo? lo está solo en sus contornos generales para realizar la investigación acerca del diseño y construcción de un brazo robótico tipo scara que puede jugar damas contra un ser humano y reconocer los movimientos realizados por el contrincante por medio de la visión artificial, se acudirá a fuentes bibliográficas, libros especializados, proyectos de tesis, revista, sitios web aquellos que contenga información acerca de la investigación.

1.7. Pasos para el procedimiento de investigación

- Identificación y selección de información.
- Localización y obtención de información.
- Consulta de información.
- Extracción de información.
- Diseño, construcción y control del brazo robótico tipo scara.
- Desarrollo del algoritmo de búsqueda, evaluación y toma de decisiones para el juego de damas.
- Reconocimiento del tablero y sus elementos por medio de visión artificial.
- Integración de toda la aplicación.

1.8. Técnicas de recolección

Las técnicas de recolección que se manejarán para la investigación del proyecto son los siguientes:

- Entrevistas personales con el director del trabajo de titulación e ingenieros de la universidad que tengan conocimiento en las áreas de robótica, inteligencia y visión artificial, para obtener información referente a las posibles soluciones que puedan encaminar a cumplir los objetivos anteriormente propuestos.

- Observar con detenimiento proyectos, prototipos, tesis y videos que contengan información de visual de los temas de estudio en nuestro proyecto y poder evidenciar en una situación real las variables que van a intervenir en diseño y construcción del prototipo.
- Experimentar con el prototipo en un ambiente para tener control sobre el entorno y así obtener los datos de las variables internas y externas que actúan sobre el prototipo y con cada nueva prueba recolectar datos que vayan mejorando las configuraciones y eliminen futuras complicaciones.

CAPÍTULO 2

VISIÓN ARTIFICIAL

2.1. Generalidades

La visión artificial es una técnica basada en la adquisición de imágenes que generalmente se encuentran en dos dimensiones, para luego procesarlas digitalmente mediante algún tipo de CPU (computadora, microcontrolador, DSP, etc.) con el fin de extraer y medir determinadas propiedades de la imágenes adquiridas. Por lo tanto, se trata de una tecnología que combina las computadoras con las cámaras de video para adquirir, analizar e interpretar imágenes de una forma equivalente a la inspección visual humana. (Universidad Nacional de Quilmes, 2005, pág. 1)

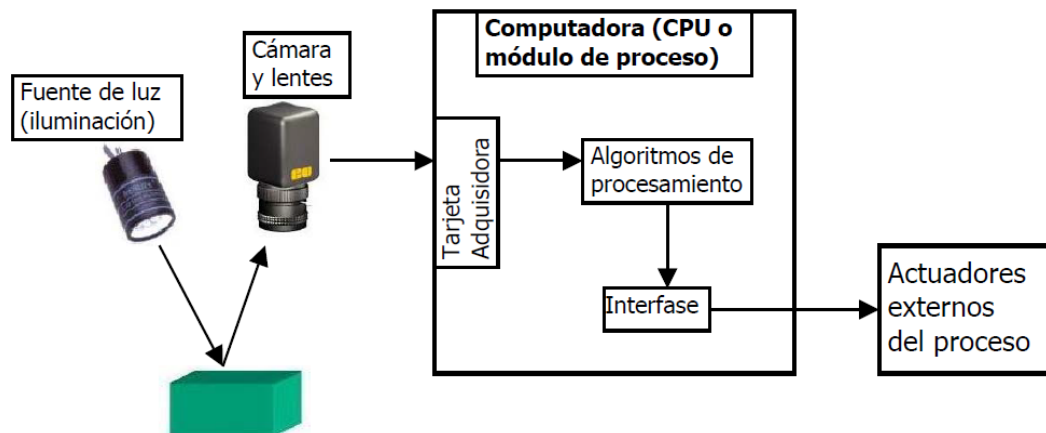
Los objetivos básicos de la visión artificial contemplan:

- La detección, segmentación, localización y reconocimiento de ciertos objetos en imágenes (por ejemplo, caras humanas).
- La evaluación de los resultados.
- Registro de diferentes imágenes de una misma escena para hacer concordar un mismo objeto en diversas imágenes.
- Seguimiento de un objeto en una secuencia de imágenes.
- Mapeo de una escena para generar un modelo tridimensional de la escena, tal modelo podría ser usado por un robot para navegar por la escena.
- Estimación de las posturas tridimensionales de humanos (Kinect).
- Búsqueda de imágenes digitales por su contenido.

Estos objetivos se consiguen por medio de reconocimiento de patrones, aprendizaje estadístico, geometría de proyección, procesamiento de imágenes, teoría de gráficos y otros campos. La visión artificial cognitiva está muy relacionada con la psicología cognitiva y la computación biológica.

2.2. Elementos básicos de un sistema de visión artificial

Figura 1. Elementos básicos que componen un sistema de visión artificial



Fuente: (Universidad Nacional de Quilmes, 2005, pág. 2)

- **Fuente de luz:** Es un aspecto de vital importancia por lo que se debe proporcionar condiciones de iluminación uniforme e independiente al entorno, facilitando en lo posible la extracción de características de interés para una determinada aplicación.
- **Sensor de imagen:** Es el encargado de recoger las características del objeto de análisis.
- **Tarjeta de captura o adquisición de imágenes:** Es la interfaz entre los sensores y la computadora o los módulos de procesamiento que permite disponer de la información capturada por los sensores.
- **Algoritmo de análisis de imágenes:** Es la parte inteligente del sistema. Su misión consiste en aplicar los algoritmos necesarios y la extracción de la información de las imágenes capturadas, con el fin de obtener los resultados deseados.
- **Computadora o módulo de proceso:** Es el sistema que analiza las imágenes recibidas por el sensor para extraer la información de interés en cada uno de los casos implementados y ejecutando los algoritmos diseñados para la obtención de los objetivos.

➤ **Sistema de respuesta en tiempo real:** Con la información adquirida, los sistemas de visión artificial pueden tomar decisiones que afecten a los sistemas físicos.

2.3. Iluminación

Es un componente vital de un sistema de visión artificial e influye directamente en la etapa de procesamiento de la imagen porque determina los filtros que se deban usar en el entorno, la consecuencia de poca iluminación va a ser que se obtenga una imagen con bajo contraste, sombras, ruido y una excesiva iluminación produce que los objetos pierdan sus colores reales, imágenes con mucho brillo que no permitan distinguir los bordes y textura.

Un buen sistema de iluminación va a permitir crear un entorno independiente y resaltar características propias del objeto de estudio.

Aspectos a tomar en cuenta para crear un sistema de iluminación:

- ¿Qué colores hay que resaltar en nuestro entorno?
- ¿Cuál es el periodo de tiempo entre la captura de cada imagen?
- ¿Cuál es la dimensión del área a iluminar?
- ¿Existen objetos que reflejen la luz?
- ¿Cuál será el fondo del escenario que permitirá distinguir o aislar los objetos a procesar?
- ¿Cuáles son las características propias del objeto que podamos resaltar?

2.3.1. Tipos de reflexión.

La luz es reflejada por los objetos de dos maneras: reflexión especular y reflexión difusa. En la reflexión especular, cada rayo incidente se refleja en una sola dirección (los rayos reflejados son paralelos). Una pista de estaño en un circuito o un espejo exhiben reflexión especular. Por otro lado, en la reflexión difusa los rayos incidentes

son dispersados en un rango de ángulos salientes. Un pedazo de papel es un reflector difuso. (Universidad Nacional de Quilmes, 2005, pág. 2)

En realidad, los objetos exhiben todas las clases de conductas entre los extremos de la reflexión especular y la difusa. Los rayos de luz reflejados por una superficie pulida de metal tienen una componente direccional dominante (reflexión especular) pero que es difundida parcialmente por irregularidades de la superficie (reflexión difusa). El papel presenta algunas propiedades especulares, como se puede comprobar al leer con luz muy intensa. (Universidad Nacional de Quilmes, 2005, pág. 2)

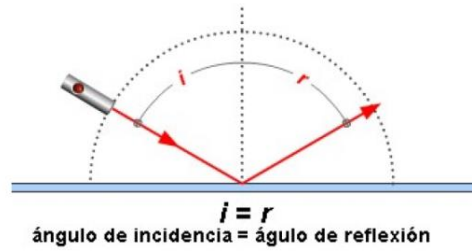
2.3.1.1. Reflexión especular.

La reflexión especular es responsable de lo que ves en el espejo cada mañana. Se lleva a cabo en superficies lisas. Cuando un rayo de luz incide sobre una superficie lisa, cada rayo en el haz se refleja con el mismo ángulo, la reflexión para especular está libre de distorsión. Puedes predecir el ángulo simplemente dibujando una línea imaginaria perpendicular a la superficie. En la figura 2 se observa, que el ángulo del rayo entrante produce una línea que es igual al ángulo del rayo reflejado; este principio se llama la ley de la reflexión. (Brennan, 2014)

2.3.1.2. Reflexión difusa.

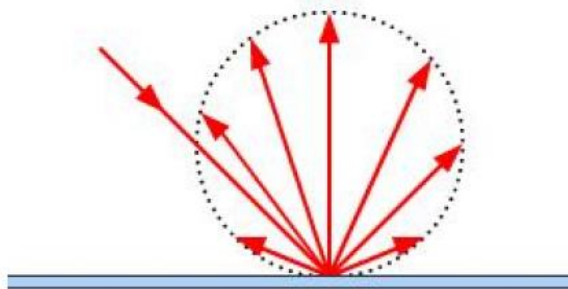
Si se pudiera mirar a los objetos a su alrededor a nivel microscópico, descubrirá que pocos, muy pocos son suaves. Los objetos tales como ropa, papel o piel humana tienen superficies ásperas, complejas. Cuando los rayos de luz tocan estas superficies, todavía obedecen la ley de la reflexión, pero la superficie es tan desigual que la línea perpendicular imaginaria ahora apunta en diferentes direcciones, dependiendo dónde la luz incide en realidad. En la figura 3 se observa que la línea del rayo entrante al ser reflejado sobre una superficie irregular, genera diferentes rayos de luz que se encuentran dispersos en muchas direcciones diferentes. Este tipo de reflexión se llama reflexión difusa. (Brennan, 2014).

Figura 2. Reflexión especular



Fuente: (Universidad Nacional de Quilmes, 2005, pág. 3)

Figura 3. Reflexión difusa



Fuente: (Universidad Nacional de Quilmes, 2005, pág. 3)

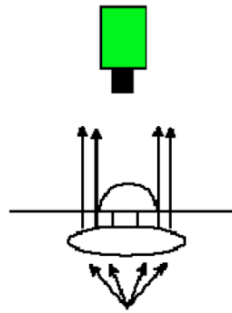
2.3.2. Técnicas de iluminación.

Existen variedad de técnicas de iluminación para resaltar los atributos de los objetos que se van a analizar, a continuación se explica los que más se utilizan en sistemas de visión artificial y que se puedan aplicar en el proyecto.

2.3.2.1. Iluminación posterior (*backlight*).

En la figura 4 se observa que la cámara se posiciona mirando al objeto desde arriba, mientras que la dirección de la luz está a bajo del objeto, lo cual visualiza el contorno de las piezas, minimiza la visibilidad de objetos traslúcidos y permite visualizar perforaciones o fisuras en objetos transparentes. Esta técnica brinda un contraste muy alto y es fácil de implementar.

Figura 4. Iluminación posterior



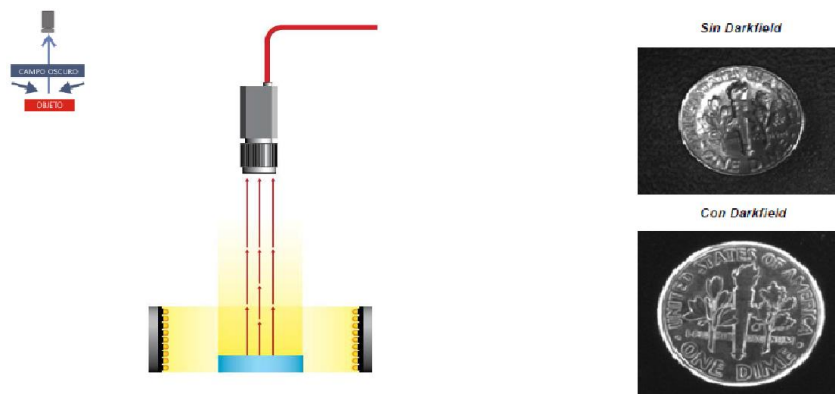
Fuente: (Universidad Nacional de Quilmes, 2005, pág. 4)

2.3.2.2. Iluminación campo oscuro.

En la figura 5 se observa la luz que incide sobre el objeto tiene un ángulo muy pequeño respecto a la superficie donde descansa. De esta manera, se consigue que sobre superficies que presentan hendiduras o elevaciones, éstas interfieran en la trayectoria de la luz produciéndose zonas brillantes, por lo cual realza el contraste de los detalles superficiales tales como grabados, marcas y defectos.

Se utiliza para resaltar incrustaciones y códigos alfanuméricos con poco contraste en metal sobre metal o gris, aplicada en la verificación de grabados tipo láser o troquel, destaca los detalles en superficies con muy poco contraste, no es recomendable en superficies que absorban la luz. (visionartificial.fpcat.cat, 2012, pág. 11).

Figura 5. Iluminación campo oscuro



Fuente: (visionartificial.fpcat.cat, 2012, pág. 11)

2.3.2.3. Iluminación frontal direccional.

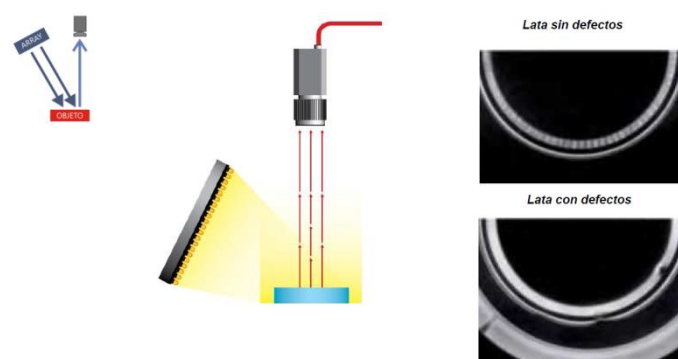
En la figura 6 se observa que la cámara se posiciona mirando al objeto mientras que la dirección de la luz es lateral al objeto. El grado de inclinación del elemento emisor de luz vendrá determinado por el grado deseado de resalte de los relieves, es muy útil para resaltar bordes, rayas y fisuras en una dirección determinada, resalta los relieves por pequeños que sean de los objetos, resultando una sombra muy definida, existen problemas con ángulos pequeños respecto a la horizontal, la luz producirá sombras en todos los relieves y en el contorno de la pieza. (visionartificial.fpcat.cat, 2012, pág. 10)

2.3.2.4. Iluminación frontal.

En la figura 7 se observa que la cámara se posiciona mirando al objeto en la misma dirección que la luz. Esto reduce las sombras, suaviza las texturas y minimiza la influencia de rayas, polvo e imperfecciones que pueda tener el objeto. La cámara recibe la luz reflejada del objeto. Este tipo de iluminación se consigue mediante anillos de luz.

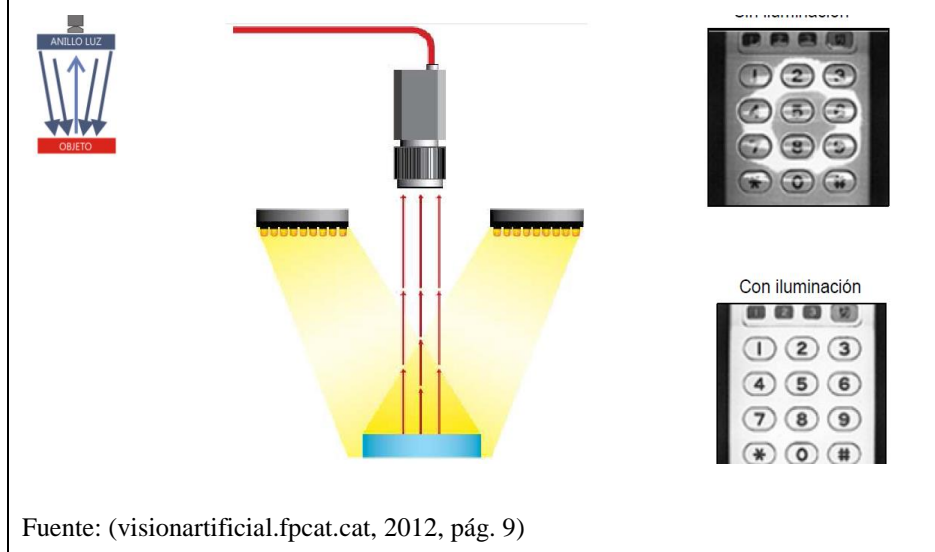
Es útil para superficies con pocos reflejos por ejemplo: papel, tela, para la detección de textura en diferentes colores, caracteres y detección de todo lo que suponga un cambio de color en prácticamente cualquier superficie, este tipo de iluminación presenta inconvenientes de intensos reflejos sobre superficies reflectante por ejemplo: metal, vidrio. (visionartificial.fpcat.cat, 2012, pág. 9)

Figura 6. Iluminación frontal direccional



Fuente: (visionartificial.fpcat.cat, 2012, pág. 10)

Figura 7. Iluminación frontal



2.3.3. Fuentes de luz.

En la figura 8 se observa los dispositivos comerciales que más se utilizan como Fuentes de luz, cada uno de ellos tiene sus ventajas y desventajas, hay que tomar en cuenta los siguientes factores:

- Intensidad lumínica.
- Durabilidad.
- Flexibilidad de diseño.
- Precio.

Los sistemas de iluminación para aplicaciones industriales pueden emitir luz de forma continua o de forma pulsada (estroboscópica).

➤ **Halógenas**

Ventajas: Muy luminosas, se puede ajustar la luminosidad.

Desventajas: No se puede utilizar como luz estroboscópica, generan mucho calor y se degradan con el tiempo.

Nota: Hay que regular la intensidad lumínica al 80% y mantener la tensión constante.

➤ **Incandescencia**

Ventajas: Precio económico.

Desventajas: No se puede utilizar como luz estroboscópica, generan calor y se degrada con el tiempo

➤ **Fluorescentes**

Ventajas: Multitud de formas y tamaños, iluminación difusa y minimiza los reflejos.

Desventajas: No se puede utilizar como luz estroboscópica, genera un parpadeo difícil de eliminar y se degrada con el tiempo.

Nota: Utiliza balastos de alta frecuencia para reducir el parpadeo.

➤ **Laser**

Ventajas: Se puede utilizar como luz estroboscópica, se puede dirigir la luz, útil para sistemas de VA en 3D.

Desventajas: Costos elevados, son muy frágiles y puede producir daños en la salud ocular para los trabajadores si se expone directamente.

➤ **Xenón**

Ventajas: Se puede utilizar como luz estroboscópica, de alta intensidad en períodos cortos (flash).

Desventajas: Se degradan con el tiempo.

Nota: Las luces estroboscópicas suelen ser de xenón.

➤ **LED (Light Emitting Diode)**

Ventaja: De larga duración (10.000 horas o más), se puede utilizar como luz estroboscópica, disponible en longitudes de onda específicas (colores) y bajo consumo de energía.

Desventajas: Los costos, pero por el tiempo de vida útil es equipara con el costo de las otras Fuentes de luz más económicas.

Nota: Actualmente la iluminación LED es la más utilizada.

2.4. Dispositivos de captura y de adquisición de datos

Los principales dispositivos que se usan para capturar imágenes son cámaras y escáneres que se pueden encontrar en el mercado, se va a centrar en análisis del funcionamiento de una webcam porque es el dispositivo a utilizar en el proyecto.

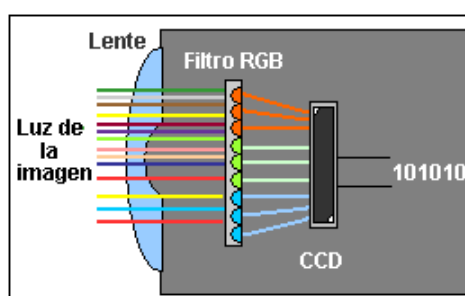
2.4.1. Webcam.

Es un periférico de entrada, en un sistema de visión artificial tiene la función de capturar y enviar imágenes o video del objeto de estudio al computador para ser procesadas por el sistema.

2.4.1.1. Cámara digital.

Originalmente estos dispositivos registran únicamente la intensidad de luz incidente, una solución para conseguir una imagen en color consiste en cubrir la retícula de celdas con filtros que sólo permitan el paso de cada una de las componentes RGB², como se observa en la figura 9, al tener 3 componentes de color aparece un problema geométrico derivado de tener que teselar³ el plano mediante grupos de 3 receptores por cada píxel, por ello estos receptores se disponen en grupos de cuatro pues la teselación es más sencilla (disposición Bayer ver Figura 9. Esquema de un CCD que utiliza un filtro Bayer), cada píxel recibe la información de cuatro receptores (RGBG), disponiendo de paso dos filtros para el verde para emular la mayor sensibilidad del ojo humano hacia ese matiz. (Vélez, Moreno, Sánchez, & Sánchez, 2007, pág. 5)

Figura 8. Funcionamiento interno de una cámara digital



Fuente: (IDLServicios.com, 2014)

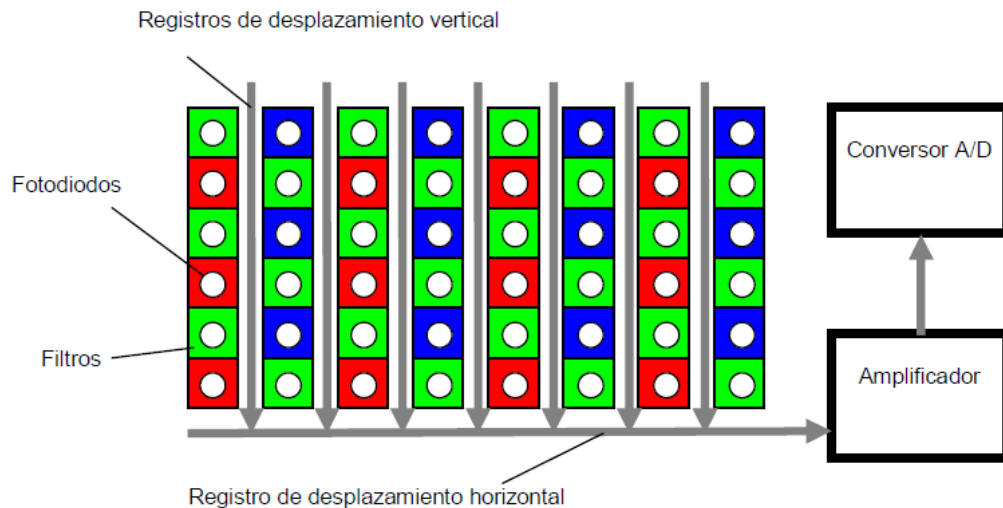
Actualmente existen dos tecnologías para estos dispositivos: C-Mos y CCD. La tecnología C-Mos es más barata al basarse en elementos semiconductores clásicos.

² Red-Green-Blue

³ Una regularidad o patrón de figuras que cubre

Sin embargo, la resolución y la respuesta conseguida con la tecnología CCD ha sido hasta ahora mejor. Este panorama puede cambiar con nuevos avances tecnológicos como el introducido por el sistema Foveon, el cual, basado en la tecnología C-Mos permite incorporar los tres receptores (R, G y B) en la misma posición física mediante un sistema multicapa. (Vélez, Moreno, Sánchez, & Sánchez, 2007, pág. 42)

Figura 9. Esquema de un CCD que utiliza un filtro Bayer



Fuente: (Vélez, Moreno, Sánchez, & Sánchez, 2007, pág. 42)

2.4.1.2. Software.

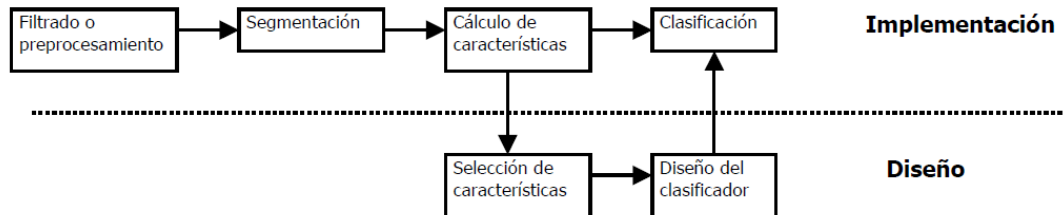
Una webcam no tiene nada de diferente con el resto de cámaras digitales y lo que le da su nombre es el software que permite la captura de las imágenes, esas capturas se llaman “frames” y ocurren cada cierto tiempo⁴ para luego ser enviadas al punto en donde son visualizadas para obtener un video de calidad, es conveniente que la webcam a utilizar pueda capturar de 15 a 30 frames por segundo de esa manera la imagen se verá más natural, si captura menos frames la imagen se ve con saltos o con movimientos muy interrumpidos. (Ministerio de Educación Cultura y Deporte de España, 2005, pág 1).

⁴ Puede ser una imagen estática en una fracción de segundo

2.5. Algoritmos de análisis de imágenes y reconocimientos de formas

Los algoritmos que se utilizan en visión artificial son muy variados e implican una gran variedad de técnicas, por esta razón a continuación se van a describir únicamente las técnicas que se utilizaron para el desarrollo del proyecto.

Figura 10. Pasos de un proceso de análisis y procesamiento de imágenes



Fuente: (Universidad Nacional de Quilmes, 2005, pág. 5)

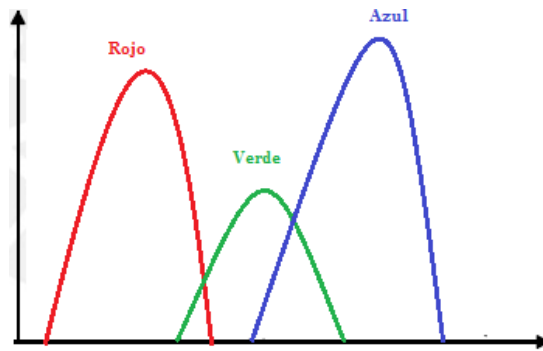
2.5.1. Filtrado o pre-procesamiento.

Cuando se adquiere una imagen mediante cualquier sistema de captura, por lo general esta no es directamente utilizable por el sistema de visión. La aparición de variaciones en intensidad debidas al ruido o por deficiencias en la iluminación o la obtención de imágenes de bajo contraste; hace necesario un pre-procesamiento de la imagen con el objetivo fundamental de corregir estos problemas, además, de aplicar aquellas transformaciones a la imagen que acentúen las características que se deseen extraer de las mismas de manera que se facilite las operaciones en las etapas posteriores.

2.5.1.1. Conversión de una imagen a color a escala de grises.

Como se sabe el ojo percibe distintas intensidades de luz en función del color que se observe, esto es debido a la respuesta del ojo al espectro visible la cual se puede observar en la figura 12, por esa razón el cálculo de la escala de grises o luminancia de la imagen debe realizarse como una media ponderada de las distintas componentes de color de cada pixel. (Sobrado, 2003, pág. 22).

Figura 11. Componentes RGB de una imagen



Elaborado por: Edgar Changoluisa

La expresión matemática de la luminancia se expresa en una ecuación 1. Los factores de ponderación de cada componente de color indican la sensibilidad del ojo humano a la frecuencia del espectro cerca al rojo, verde y azul.

Ecuación 1. *Expresión matemática de la iluminación*

$$Y = R * 0.3 + G * 0.5 + B * 0.11$$

Fuente: (Esqueda & Palafox, 2005, pág. 19)

2.5.1.2. Operaciones aritméticas de imágenes – Diferencia.

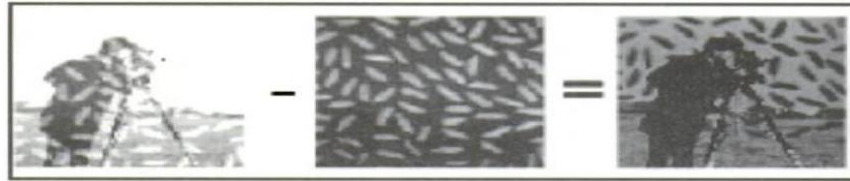
Consiste en restar de una imagen los valores de los píxeles correspondientes de otra imagen. Esta operación es un paso intermedio en algunos procesamiento más complejos como la detección de movimiento. La resta, al igual que la suma de imágenes requiere que ambas imágenes sean de igual tamaño. En la figura 13 a la izquierda la imagen original, en el centro la imagen a restar y a la derecha se muestra el efecto de restar una imagen de otra.

Ecuación 2. *Operaciones aritméticas de imágenes - diferencia*

$$C_{(x,y)} = A_{(x,y)} - B_{(x,y)}$$

Fuente: (Esqueda & Palafox, 2005, pág. 19)

Figura 12. Resta de dos imágenes



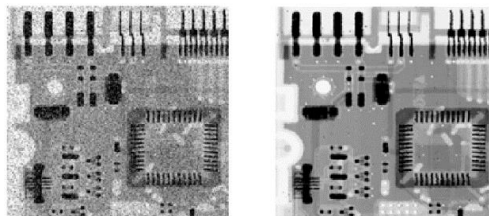
Fuente: (Esqueda & Palafox, 2005, pág. 20)

2.5.1.3. Filtro de suavizado: mediana.

El filtrado de suavizado espacial se basa en el promediado de los píxeles adyacentes al píxel que se evalúa. El filtro de mediana es muy usado debido a que proveen una excelente reducción de ruidos de tipo sal y pimienta que son los puntos negros o blancos que aparecen sobrepuestos en la imagen.

El procedimiento general para realizar el filtro de mediana en cualquier punto consiste en ordenar los valores de dicho píxel y los de su vecindario, determinar la mediana y asignar éste último valor al píxel en cuestión. Para un vecindario de 3x3, la mediana es el 5to valor más grande, para uno de 5x5 es el 13mo valor más grande, y así. Cuando algunos valores en un vecindario son iguales, éstos se ven agrupados. Por ejemplo, supongamos que un vecindario de 3x3 tiene los valores {10, 20, 20, 20, 15, 20, 20, 25, 100}. Ordenando este conjunto queda {10, 15, 20, 20, 20, 20, 20, 25, 100}, de donde se ve que la mediana es 20. Así, la principal función de los filtros de mediana es hacer que los puntos con niveles de gris distintos sean más parecidos a los de su vecindario. (Universidad Nacional de Quilmes, 2005, pág. 5).

Figura 13. Aplicación del filtro de la mediana



Fuente: (Universidad Nacional de Quilmes, 2005, pág. 5)

A lado izquierdo se observa la imagen original con ruido de sal y pimienta, a la derecha se observa la imagen luego de aplicar un filtro de mediana 3x3. Es visible que el filtro de mediana es muy efectivo para remover ruidos de sal y pimeinta, el precio que se paga es un borreado considerable.

2.5.1.4. *Filtro de suavizado: Gaussiano.*

El filtro gaussiano se usa para emborronar⁵ imágenes y eliminar ruido. Es similar al filtro de media, pero se usa una máscara diferente modelando la función gaussiana.

Ecuación 3. *Función gausseana*

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Fuente: (Universidad de Sevilla, 2013, pág. 17)

Figura 14. Aplicación del filtro Gaussiano



Fuente: (Universidad de Sevilla, 2013, pág. 18)

En la figura 15 se observa a la izquierda la imagen con ruido y a la derecha observamos la imagen luego de aplicar el filtro gaussiano con una mascara 5x5 y Sigma 4.

⁵ Llenar de borrones o garrapatos un papel (Diccionario de la real academia española)

Las ventajas del filtro gaussiano frente al filtro de media son:

- Podemos realizar dos convoluciones unidimensionales ⁶. Una en sentido horizontal y otra en sentido vertical.
- Produce un suavizado más uniforme.

2.5.2. Segmentación o aislamiento de los objetos de interés.

Cuando se dispone de la imagen capturada y filtrada, es necesario aislar o separar los objetos de interés de la escena. Se pretende detectar automáticamente los bordes entre los elementos o regiones. Las operaciones de segmentación de una escena dependen de la propia escena y de la información que se busque dentro de la imagen (Universidad Nacional de Quilmes, 2005, pág. 6). Por ejemplo:

- En fotografía aérea se utiliza para separar regiones pobladas, agrícolas, bosques, aguas, etc.
- En imagen médica se usa para aislar zonas enfermas de los órganos o para detectar y contar microorganismos en microscopia.

2.5.2.1. Algoritmo de Canny.

El algoritmo de Canny es un operador desarrollado por John F. Canny en 1986 que utiliza un algoritmo de múltiples etapas para detectar una amplia gama de bordes en imágenes. El propósito de Canny era descubrir el algoritmo óptimo de detección de bordes. Para que un detector de bordes pueda ser considerado óptimo debe cumplir los siguientes puntos: (Opencv dev team, 2013)

- **Buena detección:** El algoritmo debe marcar el mayor número real en los bordes de la imagen como sea posible.

⁶ Se denomina convolución a una función, que de forma lineal y continua, transforma una señal de entrada en una nueva señal de salida

➤ **Buena localización:** Los bordes de marca deben estar lo más cerca posible del borde de la imagen real.

➤ **Respuesta mínima:** El borde de una imagen sólo debe ser marcado una vez y siempre que sea posible; el ruido de la imagen no debe crear.

Etapas del algoritmo de Canny:

➤ **Reducción de ruido:** El algoritmo de detección de bordes de Canny utiliza un filtro basado en la primera derivada de una gaussiana. Ya que es susceptible al ruido presente en datos de imagen sin procesar, la imagen original es transformada con un filtro gaussiano. El resultado es una imagen un poco borrosa respecto a la versión original. Esta nueva imagen no se ve afectada por un píxel único de ruido en un grado significativo.

Ecuación 4. *Ejemplo de un filtro gaussiano 5x5 y $\sigma = 1.4$*

$$K = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Fuente: (opencv dev team, 2013)

➤ **Encontrar la intensidad del gradiente de la imagen:** El borde de una imagen puede apuntar en diferentes direcciones, por lo que el algoritmo de Canny utiliza cuatro filtros para detectar horizontal, vertical y diagonal en los bordes de la imagen borrosa. El operador de detección de bordes devuelve un valor para la primera derivada en la dirección horizontal (G_y) y la dirección vertical (G_x). A partir de éste, se pueden determinar el gradiente de borde y la dirección.

Ecuación 5. *Ecuaciones para encontrar gradiente y dirección de la imagen*

$$G = \sqrt{G_x^2 + G_y^2}$$
$$\Theta = \arctan\left(\frac{G_y}{G_x}\right)$$

Fuente: (opencv dev team, 2013)

2.5.2.2. Dilatación.

Sean A y B conjuntos en Z^2 . La dilatación de A por B , expresada por $A \oplus B$, se define como:

Ecuación 6. *Ecuación de dilatación (1)*

$$A \oplus B = \{Z \mid (B^{\wedge})_z \cap A \neq \emptyset\}$$

Fuente: (opencv dev team, 2013)

Esta ecuación consiste en obtener la reflexión de B sobre su origen y trasladar esta reflexión por z . La dilatación de A por B es entonces el conjunto de todos los desplazamientos z , tal que la reflexión de B y A se solapan por al menos un elemento. Teniendo en cuenta lo anterior, la dilatación de A por B también se puede expresar como:

Ecuación 7. *Ecuación de dilatación (2)*

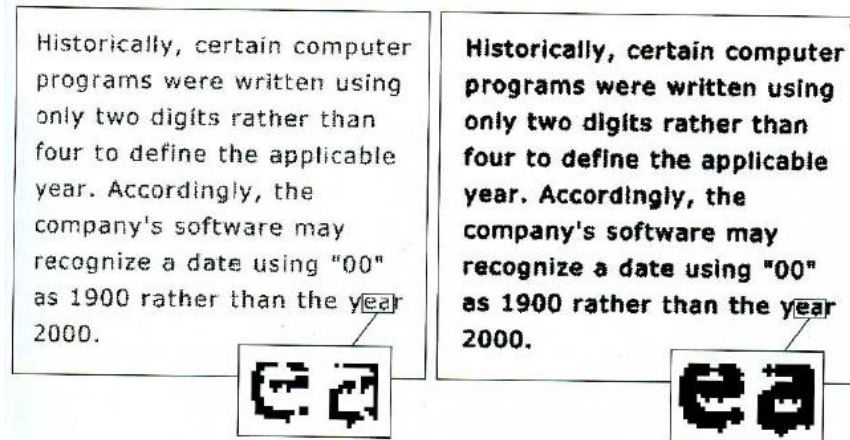
$$A \oplus B = \{z \mid ((B^{\wedge})_z \cap A) \subseteq A\}$$

Fuente: (opencv dev team, 2013)

En general, la dilatación aumenta el tamaño de un objeto. La cantidad y la forma en que aumenta el tamaño dependen de la elección del elemento estructural.

Una de las aplicaciones más simples de la dilatación es la unión de píxeles relacionados. La figura 16 (a) muestra un texto cuyos caracteres han perdido píxeles debido a un filtrado con pasábandos. Se dilata la imagen y se pueden unir los caracteres partidos y obtener la figura 16 (b) (Universidad Nacional de Quilmes, 2005, pág. 5)

Figura 15. Ejemplo de dilatación de bordes de una imagen.

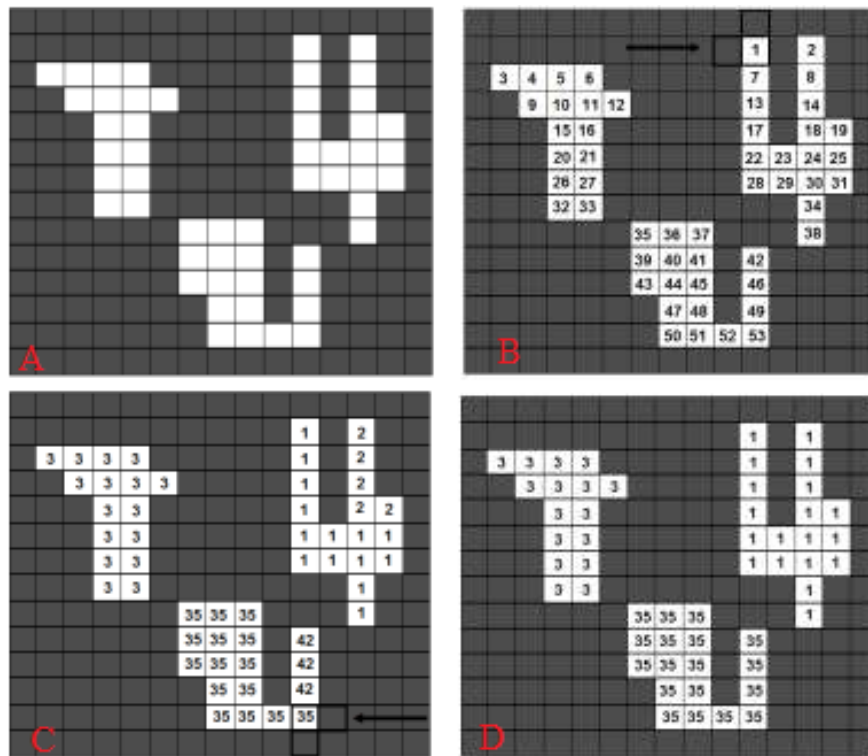


Fuente: (Universidad Nacional de Quilmes, 2005, pág. 6)

2.5.2.3. *Etiquetado.*

Para poder identificar los píxeles que pertenecen a cada una de las regiones y asociar estas regiones a los objetos para extraer sus características, es necesario obtener las componentes conexas presentes en la imagen. Si en la imagen aparecen varias regiones no se podrá obtener, por ejemplo, su área o centro de gravedad hasta que estas no estén identificadas de forma individual. Este proceso se conoce como etiquetado y consiste en definitiva en asignar una misma etiqueta a cada uno de los píxeles que pertenecen a una componente conexa. La etiqueta será un valor numérico que permite identificar todos los píxeles que forman parte de una misma región. De esta forma, la imagen de etiquetas obtenida permitirá la extracción individualizada de características de cada una de las regiones conexas de la imagen. (Fuente & Trespaderne, 2012, pág. 69).

Figura 16. Procesamiento de etiquetado



Fuente: (Fuente & Trespaderne, 2012, pág. 72)

En la figura 17 se observa la imagen (a) la cual se va a etiquetar. Como primer paso vamos a inicializar numerando todos los píxeles blancos (b). A continuación, haciendo un barrido de arriba abajo y de izquierda a derecha se actualiza cada píxel si su etiqueta es mayor que el valor mínimo de etiqueta que existe en su vecindad. En la figura se ha considerado 4-adyacencia. (c) Finalizado el barrido de arriba abajo y de izquierda a derecha se procede análogamente pero de abajo a arriba y de derecha a izquierda (d) imagen obtenida. Cada objeto que aparecía en la imagen binaria en blanco se muestra ahora con una etiqueta exclusiva. Debe tenerse en cuenta que este algoritmo necesita un proceso que reasigne las etiquetas ya que las obtenidas no son consecutivas. Por ejemplo, en esta imagen con tres objetos, un objeto tiene la etiqueta 1, otro la 3 y otro la 35 (Fuente & Trespaderne, 2012, pág. 72).

CAPÍTULO 3

INTELIGENCIA ARTIFICIAL , ALGORITMOS PARA JUEGOS

3.1. Introducción

Desde el principio de los tiempos el hombre se ha preguntado que es la inteligencia. Los hombres se han denominado a sí mismos como homo sapiens⁷ porque nuestras capacidades mentales son muy importantes para nosotros. Durante miles de años, se a tratado de entender como se piensa; es decir, entender como un simple puñado de materia puede percibir, entender, predecir y manipular un mundo mucho mas grande y complicado que ella misma. El campo de la inteligencia artificial va más allá, no solo intenta comprender sino también se esfuerza en construir entidades inteligentes (Russell & Norvig, 2004, pág. 2).

De acuerdo al libro “Inteligencia Artificial , un enfoque moderno” de Stuart J. Russell y Peter Norvig, La inteligencia artificial (IA) es una de las ciencias más recientes. El trabajo comenzó poco después de la segunda guerra mundial y el nombre se acuño en 1956. Sin embargo, el estudio de la inteligencia contemplada como el razonamiento humano viene siendo estudiado por los filósofos hace más de dos milenios.

La inteligencia artificial abarca en la actualidad una gran variedad de sub-campos que van desde áreas de propósito general como el aprendizaje y la percepción, a otras más específicas como la resolución de juegos de lógica y matemática como el ajedrez, la demostración de teoremas matemáticos, la escritura y el diagnóstico de enfermedades. La inteligencia artificial sintetiza tareas intelectuales y es por lo tanto potencialmente relevante para cualquier ámbito de la actividad intelectual humana. Mirando desde esta perspectiva es un campo genuinamente universal (Russell & Norvig, 2004, pág. 78)

⁷ Hombre sabio

En este capítulo se revisará algunas definiciones generales de inteligencia y posteriormente intentaremos definir que es inteligencia artificial. Para poder cumplir con los objetivos de este proyecto se limitará al estudio de la IA aplicada en la resolución de juegos.

El avance en el estudio de los algoritmos para juegos, búsquedas, evaluaciones y adicionalmente el incremento progresivo de la potencia de las computadoras han hecho prácticamente imposible que un ser humano pueda ganar en juegos de lógica y matemática como el ajedrez o las damas.

3.2. ¿Qué es la inteligencia?

La introducción a este capítulo hace una descripción de lo importante que hoy en día es esta ciencia para la vida humana, pero no se ha explicado: ¿Qué es? Para lo cual en la tabla 1 se presenta las definiciones de la inteligencia artificial extraídas de ocho libros de texto.

Tabla 1. *Definición de inteligencia artificial, organizada en 4 categorías.*

| Sistemas que piensan como humanos | Sistemas que piensan racionalmente |
|---|--|
| «El nuevo y excitante esfuerzo de hacer que los computadores piensen... máquinas con mentes, en el más amplio sentido literal». (Haugeland, 1985) | «El estudio de las facultades mentales mediante el uso de modelos computacionales». (Charniak y McDermott, 1985) |
| «[La automatización de] actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolución de problemas, aprendizaje...» (Bellman, 1978) | «El estudio de los cálculos que hacen posible percibir, razonar y actuar». (Winston, 1992) |
| Sistemas que actúan como humanos | Sistemas que actúan racionalmente |
| «El arte de desarrollar máquinas con capacidad para realizar funciones que cuando son realizadas por personas requieren de inteligencia». (Kurzweil, 1990) | «La Inteligencia Computacional es el estudio del diseño de agentes inteligentes». (Poole <i>et al.</i> , 1998) |
| «El estudio de cómo lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor». (Rich y Knight, 1991) | «IA... está relacionada con conductas inteligentes en artefactos». (Nilsson, 1998) |

Fuente: (Russell & Norvig, 2004, pág. 154)

En la tabla 1 las definiciones que se observan hacen referencia los procesos mentales y razonamientos del ser humano, mientras que las que se observan en la parte inferior

apuntan a la conducta. Las definiciones de la izquierda miden el éxito en términos de la fidelidad en la forma de actuar de los humanos, mientras que las de la derecha toman como referencia un concepto ideal de inteligencia que llamaremos racionalidad. Un sistema es racional si se hace "lo correcto", en función de su conocimiento.

A lo largo de la historia se han seguido los cuatro enfoques mencionados. Como es de esperar, existe un enfrentamiento entre los enfoques centrales en los humanos y los centrados en torno a la racionalidad. El enfoque centrado en el comportamiento humano debe ser una ciencia empírica que incluya hipótesis y confirmaciones mediante experimentos. El enfoque racional implica una combinación de matemáticas e ingeniería.

Finalmente, se puede decir que los conceptos de inteligencia son relativos y varían dependiendo del tipo de área de investigación, en general están relacionados a la capacidad de generar conocimiento para aplicarlos en la solución de problemas, la inteligencia y conocimiento son conceptos que se vinculan.

3.3. ¿Qué es la inteligencia artificial?

No existe una definición concreta de inteligencia artificial porque los padres de la misma no han podido llegar a un consenso para definirla, pero la mayoría de autores coincide en que: “la inteligencia artificial es una de las áreas más fascinantes y con más retos de las ciencias de la computación, en su área de ciencias cognitivas. Nació como mero estudio filosófico y de razonamiento de la inteligencia humana mezclada con la inquietud del hombre de imitar la naturaleza circundante hasta inclusive querer imitarse a sí mismo.” (Colegio Teodoro Alvarado Olea, 2010).

3.4. Juegos

Para los seres humanos los juegos despiertan una gran fascinación, por tal motivo han ocupado gran parte de sus facultades intelectuales en ellos. Para los investigadores de inteligencia artificial la naturaleza abstracta de los juegos han hecho que sea un tema atractivo de estudio. Las situaciones del juego proporcionan entornos de problemas controlados porque existe un estado inicial, los participantes

están restringidos por un número de reglas ya definidas, pero poseen la suficiente complejidad por lo cual se necesitan habilidades de razonamiento e inteligencia para llegar a la solución.

Los juegos proporcionan una tarea estructurada en la que es muy fácil medir el éxito o el fracaso. En comparación con otras aplicaciones de inteligencia artificial, por ejemplo, comprensión del lenguaje, los juegos no necesitan grandes cantidades de conocimiento.

Las computadoras han superado a los grandes campeones en juegos como damas y en Othello, pero el ajedrez y backgammon no siempre lo han logrado debido a su dificultad para ser resueltos, por ejemplo, el juego de ajedrez, en una partida cada jugador realiza una media de 50 movimientos, con un factor de ramificación medio de 35 posibilidades, por lo tanto, para examinar el árbol de juego completamente se tendría que examinar 35^{100} posibilidades (Abásolo, 2001, pág. 1). Por lo tanto, los juegos, como el mundo real, requieren la capacidad de tomar alguna decisión cuando no se puede calcular la decisión óptima.

Para mejorar la efectividad de un programa que solucione problemas en particular de juegos es importante:

- Mejorar el algoritmo para que sólo generen movimientos que permitan ganar.
- Mejorar el algoritmo de búsqueda y toma de decisiones para que sólo se reconozcan los mejores movimientos.

En función de los ítems anteriores se incorpora la búsqueda o exploración de distintos niveles del juego para conocer que puede ocurrir más adelante. Resulta evidente que la elección del camino a seguir será más acertado mientras más niveles sean explorados antes de tomar la decisión. Se utiliza una función de evaluación estática para elegir el movimiento más prometedor. Shannon utilizó una función lineal de funciones de evaluación simples para evaluar un tablero de damas (Abásolo, 2001, pág. 1) .

Ecuación 8. *Función lineal de Shanon*

$$C_1 * \text{ventaja piezas} + C_2 * \text{avance} + C_3 * \text{amenazas dobles} + \dots$$

Fuente: (opencv dev team, 2013)

En la función anterior, con un mecanismo de aprendizaje, los pesos o ponderaciones C_i se incrementan o disminuyen siempre que sus componentes sugieran movimientos que conducen a la victoria o al fracaso respectivamente.

3.4.1. Métodos de análisis de juegos por computadora.

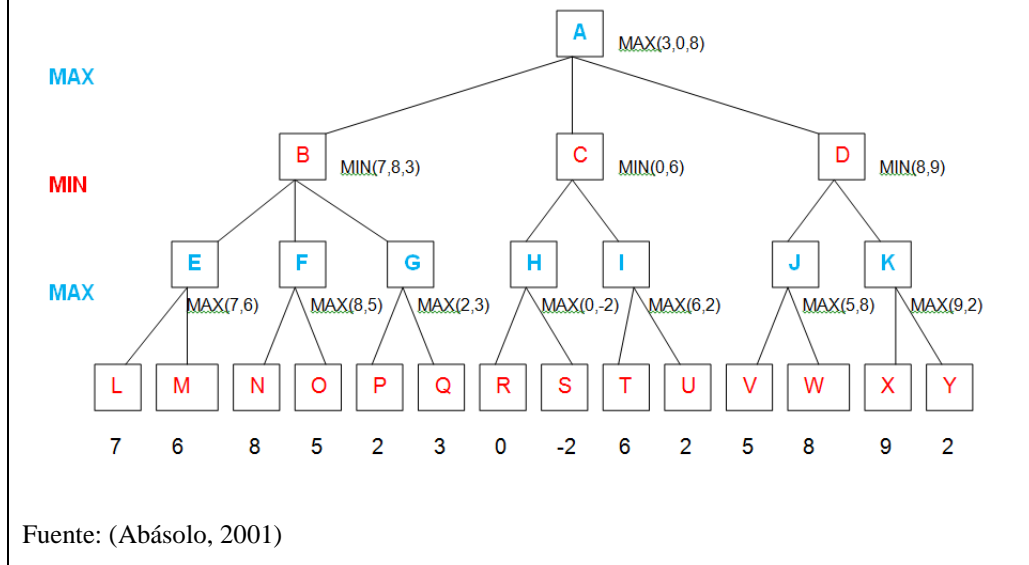
El método más conocido y que ha constituido la base para desarrollar muchas de las implementaciones de juegos inteligentes de computadora se llama minimax. A continuación, se van a describir de manera rápida los métodos utilizados en juegos de información completa como son el ajedrez y las damas.

3.4.1.1. Búsqueda minimax.

La búsqueda minimax puede resumirse como elegir el mejor movimiento para ti mismo suponiendo que tu contrincante escogerá el peor para ti. (Abásolo, 2001)

El nombre del algoritmo deriva de considerar que, dada una función estática que devuelve valores en relación al jugador max, éste procura maximizar su valor mientras que su oponente procura minimizarlo. En un árbol de juego donde los valores de la función estática están en relación al jugador max, se maximiza y minimiza alternadamente de un nivel a otro. La figura 18 se muestra un ejemplo donde la elección del siguiente movimiento según la búsqueda minimax de tres niveles sería el nodo D. (Abásolo, 2001, pág. 1).

Figura 17. Árbol de juegos minimax



El algoritmo minimax es un procedimiento recursivo y el corte de la recursión está dado por alguna de las siguientes condiciones:

- Gana algún jugador.
- Se han explorado N capas, siendo N el límite establecido.
- Se ha agotado el tiempo de exploración.
- Se ha llegado a una situación estática donde no hay grandes cambios de un nivel a otro.

En la figura 19 se observa el pseudocódigo del algoritmo para el cálculo de decisiones minimax. Devuelve la acción correspondiente al mejor movimiento posible, es decir, el movimiento que conduce al resultado con la mejor utilidad, conforme al axioma que el oponente juega para minimizar la utilidad. Las funciones valor-max menos el valor-min pasan por el árbol de juegos entero, por todos los caminos hacia las hojas, para determinar el valor que le llega a un estado (Russell & Norvig, 2004, pág. 186).

Figura 18. Algoritmo minimax

```
función DECISIÓN-MINIMAX(estado) devuelve una acción  
  variables de entrada: estado, estado actual del juego  
   $v \leftarrow \text{MAX-VALOR}(\text{estado})$   
  devolver la acción de SUCESORES(estado) con valor  $v$   


---

función MAX-VALOR(estado) devuelve un valor utilidad  
  si TEST-TERMINAL(estado) entonces devolver UTILIDAD(estado)  
   $v \leftarrow -\infty$   
  para un  $s$  en SUCESORES(estado) hacer  
     $v \leftarrow \text{MAX}(v, \text{MIN-VALOR}(s))$   
  devolver  $v$   


---

función MAX-VALOR(estado) devuelve un valor utilidad  
  si TEST-TERMINAL(estado) entonces devolver UTILIDAD(estado)  
   $v \leftarrow \infty$   
  para un  $s$  en SUCESORES(estado) hacer  
     $v \leftarrow \text{MIN}(v, \text{MAX-VALOR}(s))$   
  devolver  $v$ 
```

Fuente: (Russell & Norvig, 2004, pág. 186)

3.4.1.2. Poda alfa-beta de la búsqueda minimax.

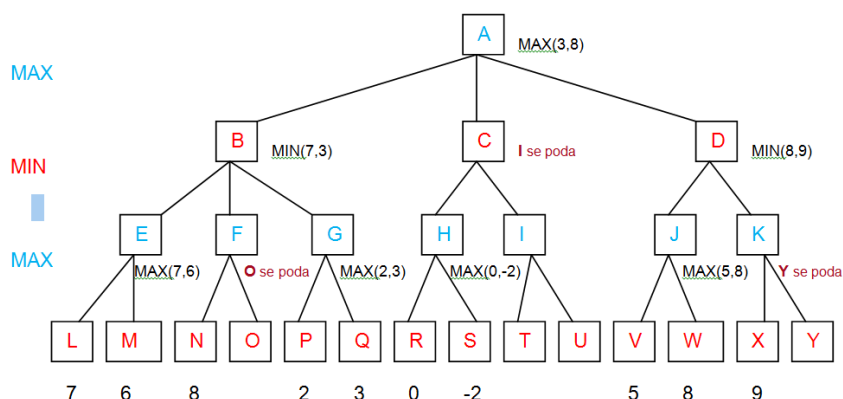
El problema de la búsqueda minimax es que el número de estados que tiene que analizarse exponencial en el número de movimientos. Lamentablemente, no se puede eliminar el exponente, pero se puede dividirlo con eficacia en la mitad. Es posible calcular la decisión de minimax sin mirar todos los nodos del árbol de juego, tomar la idea de “podar” a fin de eliminar partes grandes del árbol. La poda alfa-beta consigue su nombre de los dos parámetros que describen los límites sobre los valores que aparecen durante del camino (Russell & Norvig, 2004, pág. 188):

α = el valor más alto que hemos encontrado hasta el momento en cualquier punto elegido a lo largo del camino para max.

β = el valor más bajo que hemos encontrado hasta el momento en cualquier punto elegido a lo largo del camino para min.

La figura 20 muestra la aplicación de la poda alfa-beta, en el ejemplo de la figura 18, al evaluar el nodo E se asegura al oponente o jugador MIN un parámetro β con valor 7, es decir, el oponente obtendrá el valor $\beta \leq 7$ en la evaluación de B porque el valor más bajo es el mejor para min. A continuación, cuando se examina el nodo N cuyo valor es 8, dado que es mayor que β , los nodos hermanos de N para este caso el nodo O, pueden podarse, dado que nos hallamos en un nivel de max, el nodo F tendrá un valor $\alpha \geq 8$.

Figura 19. Aplicación de la poda alfa-beta



Fuentes: (Abásolo, 2001)

Luego de evaluar los sucesores de B, se concluye que este nodo asegura al jugador max un parámetro β con valor 3, es decir, obtendrá 3 o un valor mayor en la evaluación de A. En este caso 3 representa el valor de α . A continuación, cuando se examina el nodo H cuyo valor es 0, dado que es menor que α , los nodos hermanos de H, en este caso el nodo I, pueden podarse, el nodo C tendrá un valor menor o igual a 0, por lo cual no podrá competir con el valor α asegurado en el nivel anterior del jugador max (Abásolo, 2001, pág. 1) .

El valor de umbral alfa o beta según corresponda, debe actualizarse cuando se encuentra un umbral mejor. En el ejemplo anterior, al obtenerse el valor 8 del nodo D se puede actualizar el valor de α de 3 a 8. Esto tendría sentido en el caso de que existieran otros nodos hermanos de D que pudieran ser podados utilizando este valor de α . De la misma manera, al obtenerse un valor de 3 en el nodo G se puede actualizar el valor de β 7 a 3. Esto tendría sentido en el caso de que existieran otros

nodos hermanos G que pudieran ser podados utilizando este valor de beta (Abásolo, 2001).

Figura 20. El algoritmo de la búsqueda alfa-beta

```

función BÚSQUEDA-ALFA-BETA(estado) devuelve una acción
variables de entrada: estado, estado actual del juego

v ← MAX-VALOR(estado,  $-\infty$ ,  $+\infty$ )
devolver la acción de SUCESORES(estado) con valor v

```

```

función MAX-VALOR(estado,  $\alpha$ ,  $\beta$ ) devuelve un valor utilidad
variables de entrada: estado, estado actual del juego
 $\alpha$ , valor de la mejor alternativa para MAX a lo largo del camino a estado
 $\beta$ , valor de la mejor alternativa para MIN a lo largo del camino a estado

si TEST-TERMINAL(estado) entonces devolver UTILIDAD(estado)
v ←  $-\infty$ 
para a, s en SUCESORES(estado) hacer
  v ← MAX(v, MIN-VALOR(s,  $\alpha$ ,  $\beta$ ))
  si v ≥  $\beta$  entonces devolver v
   $\alpha$  ← MAX( $\alpha$ , v)
devolver v

```

```

función MIN-VALOR(estado,  $\alpha$ ,  $\beta$ ) devuelve un valor utilidad
variables de entrada: estado, estado actual del juego
 $\alpha$ , valor de la mejor alternativa para MAX a lo largo del camino a estado
 $\beta$ , valor de la mejor alternativa para MIN a lo largo del camino a estado

si TEST-TERMINAL(estado) entonces devolver UTILIDAD(estado)
v ←  $+\infty$ 
para a, s en SUCESORES(estado) hacer
  v ← MIN(v, MAX-VALOR(s,  $\alpha$ ,  $\beta$ ))
  si v ≤  $\alpha$  entonces devolver v
   $\beta$  ← MIN( $\beta$ , v)
devolver v

```

Fuente: (Russell & Norvig, 2004, pág. 190)

En la figura 21 se puede notar que las rutinas son las mismas del algoritmo minimax de la figura 19, excepto por las dos líneas min-valor y max-valor que permiten actualizar los valores de α y β , según se va recorriendo el árbol y poda las ramas restantes de un nodo tan pronto como el valor del nodo actual es peor que el valor actual de α o β para max o min, respectivamente (Russell & Norvig, 2004, pág. 190).

3.4.1.3. Toma de decisiones en tiempo real.

El algoritmo minimax genera el espacio de búsqueda completo, mientras que el algoritmo alfa-beta, permite eliminar partes grandes del espacio de búsqueda. Sin embargo, alfa-beta todavía tiene que buscar en todos los caminos y su efectividad depende en gran medida del orden en que se examinen los caminos. Si se examinan primero los peores caminos, no se realizará ningún corte. Esto no es práctico para

poder jugar contra un ser humano porque los movimientos deben hacerse en una cantidad razonable de tiempo es decir en unos pocos minutos como máximo (Russell & Norvig, 2004, pág. 191).

A continuación se describen otras modificaciones al algoritmo minimax y la poda Alfa-Beta que mejora su rendimiento.

3.4.1.3.1. Funciones de evaluación.

La función de evaluación, también conocida como función de evaluación heurística es un algoritmo usado generalmente por programas que saben jugar juegos de estrategia como el ajedrez, dama, GO entre otros; para estimar el valor numérico a las posiciones analizadas por el programa por medio de un algoritmo de búsqueda como la poda alfa-beta, que deberá tomar la decisión de por cual es el camino a seguir en el árbol de variantes del juego. La función de evaluación generalmente solo da información muy básica sobre la instancia del juego para no usar mucho recursos del procesador, de esta manera, la función de búsqueda puede explorar el árbol de variantes más rápidamente y por lo tanto mejorando el rendimiento del programa durante el juego.

La mayoría de las funciones de evaluación trabajan calculando varias características del estado, por ejemplo, en el juego de ajedrez, el número de peones capturados por cada lado. Las características juntas definen varias categorías o clases de equivalencia de estados, los estados en cada categoría tienen los mismos valores para todas las características. Cualquier categoría dada, por lo general, tendrá algunos estados que conducen a triunfos, algunos que conducen a empates y algunos que conducen a pérdidas. La función de evaluación no sabe cuál es cada estado, pero sí puede devolver un valor que refleje la proporción de estados con cada resultado (Russell & Norvig, 2004, pág. 192).

En la práctica esta clase de análisis requiere demasiadas categorías y demasiada experiencia para estimar todas las probabilidades de ganar, en cambio, la mayoría de las funciones de evaluación calculan las contribuciones numéricas de cada

característica y luego las combinan para encontrar el valor total. Por ejemplo para el juego de las damas utilizaremos:

Ecuación 9. *Contribuciones numéricas de cada característica*

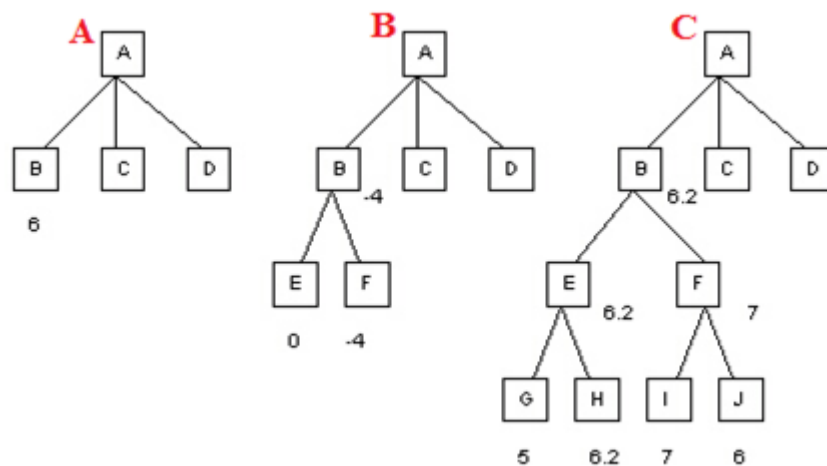
$$F=3*\text{damasMaquina} + \text{peonesMaquina} - 3*\text{damasUsuario} - \text{peonesUsuario}$$

Elaborado por: Edgar Changoluisa

3.4.1.3.2. Corte de la búsqueda.

En la poda alfa-beta también se debe llevar la contabilidad de la profundidad del árbol que se está explorando, de modo que la profundidad actual se incremente sobre cada llamada recursiva. La aproximación más sencilla para controlar la cantidad de búsqueda es poner un límite de profundidad fijo, de modo que se detenga la búsqueda cuando se supere el límite definido, él mismos se elige de modo que la cantidad de tiempo usado no exceda de lo que permiten las reglas del juego (Russell & Norvig, 2004, pág. 194).

Figura 21. Búsqueda alfa y beta con límite de profundidad fija.



Fuente: (Abásolo, 2001)

Cuando la condición de corte de la recursión de la poda alfa y beta está basada sólo en la profundidad fija del árbol explorado, puede darse el llamado efecto horizonte. Esto ocurre cuando se evalúa como buena o mala una posición, sin saber que en la siguiente jugada, la evaluación sea ficticia.

Para evitar el efecto horizonte, una de las condiciones de corte de recursión en la poda alfa y beta, debería ser el alcanzar una situación estable. Si se evalúa un nodo de un árbol de juego y este valor cambia drásticamente al evaluar el nodo después de realizar la exploración de un nivel más del árbol, la búsqueda debería continuar hasta que esto dejara de ocurrir. Esto se denomina esperar el reposo y nos asegura que las medidas a corto plazo, por ejemplo un intercambio de piezas, no influyen indebidamente en la elección (Abásolo, 2001, pág. 1).

En la figura 22 en el ítem A, se puede observar la evaluación del nodo B que tiene el valor 6 explorando sólo un nivel del árbol. Si se explora un nivel más, la evaluación da un valor de -4, como se muestra en el ítem B, Esto denota un cambio muy drástico de evaluación del nodo B. Por esto, se decide explorar un nivel más el árbol, como se observa en la ítem C, resultando en una evaluación con valor de 6.2, la cual no difiere mucho de la evaluación inicial. En este nivel podría determinarse abandonar la exploración debido a la situación de reposo alcanzada.

CAPÍTULO 4

ROBÓTICA- ROBOT SCARA

4.1. Introducción

La palabra robot, se vincula con películas de ciencia ficción que han sorprendido por presentar robots que realizan acciones superiores a las capacidades del ser humano. Sin embargo, la idea que presentan las películas se encuentra bastante alejada de la aplicación industrial de los robots, a los cuales se les considera como un tipo de máquinas herramientas. La palabra robot es de origen eslavo. En ruso significa trabajo. En checo significa trabajo forzado (Iñigo & Vidal, 2002, pág. 1).

El desarrollo de máquinas ha estado fuertemente influido por el progreso tecnológico. De esta forma, se pasa de máquinas que tiene como objetivo exclusivo la amplificación de la potencia muscular del hombre, sustituyendo su trabajo físico, a máquinas o instrumentos que son también capaces de procesar información. Complementando o incluso sustituyendo, al hombre en algunas actividades intelectuales (Ollero, 2001, pág. 1).

Desde la antigüedad, el hombre ha sentido fascinación por las máquinas que imitan la forma y movimiento de los seres vivos, existe una larga tradición de autómatas desde el mundo de los griegos hasta nuestro siglo, pasando por el siglo XVIII que los artesanos francés y suizos ya incorporan interesantes dispositivos mecánicos para el control automático de los movimientos.

El primer brazo robótico articulado fue construido por Harold Roselund de la compañía Devilviss, en 1938. Se usaba para pintar con spray forzado (Iñigo & Vidal, 2002, pág. 1). El robot industrial es el resultado de la unión de una estructura mecánica articulada y de un sistema electrónico de control, en el que se integra un procesador como “cerebro” del sistema, que permite la programación y el control de los movimientos, así como la memorización de diversas secuencias de trabajo, dotando al robot de una gran flexibilidad y posibilita su adaptación a diversos trabajos o tareas. El robot industrial se especifica y diseña en función de los diversos movimientos que debe efectuar y no a partir de operaciones concretas a ejecutar (Sotomayor , 2009, pág. 3).

En los robots industriales la programabilidad y adaptabilidad los convierte en elementos esenciales de los sistemas de fabricación flexible, en los que en contraposición a los de automatización fija que siguen una secuencia preestablecida para la fabricación de un producto, pueden alterar su protocolo para ajustarse a las demandas variables de producción. Actualmente los robots industriales, no toman decisiones en función de parámetros o factores que cambian de forma durante su funcionamiento, sin embargo los recientes progresos en los campos de la inteligencia artificial , del aprendizaje y del reconocimiento de formas permiten a los robots industriales obtener información de sus entorno y adaptar su comportamiento a las modificaciones del mismo, sin necesidad de intervención de un operario o programador (Sotomayor , 2009, pág. 3).

4.2. Definición.

La complejidad para definir el concepto de robot, alguna de las definiciones se debe a su fabricación flexible. Pero resulta difícil establecer una definición formal a continuación se presentan algunas definiciones:

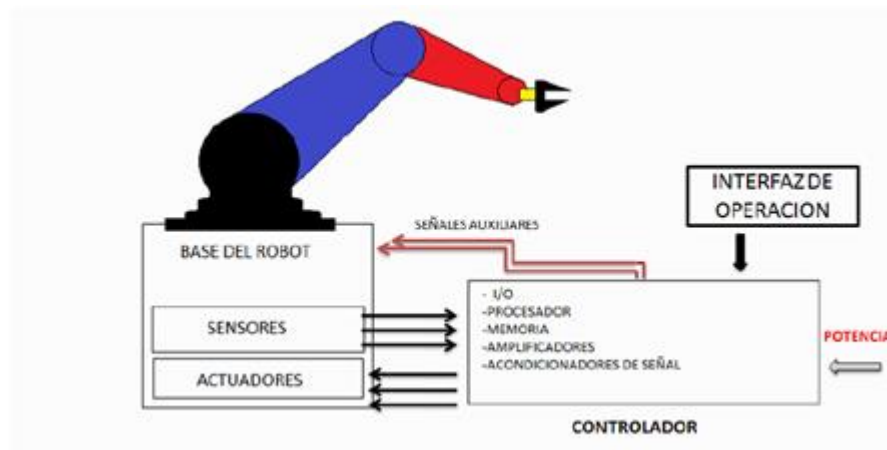
- **Japan Industrial Robots Association (JIRA):** "Es una máquina capaz de realizar movimientos versátiles parecidos a los de nuestras extremidades superiores, con cierta capacidad sensorial y de reconocimiento y capaces de controlar su comportamiento" (Martínez, 2012, pág. 3).
- **Robot Industrial Association (RIA) o Robot Institute of America:** "Es un manipulador multifuncional y reprogramable diseñado para mover cargas, materiales, piezas, herramientas y dispositivos especiales según variadas trayectorias, programados para realizar diferentes tareas" (Martínez, 2012, pág. 3).
- **Asociación internacional de estándares (ISO):** "Manipulador multifuncional reprogramable con varios grados de libertad, capaz de manipular materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas" (Martínez, 2012, pág. 3).
- **Asociación Francesa de Normalización (AFNOR):** "Manipulador automático, servo controlado, reprogramable, polivalente capaz de posicionar y orientar piezas, útiles o dispositivos especiales, siguiendo trayectorias variables

reprogramables, para la ejecución de tareas variadas. Normalmente tiene la forma de uno o varios brazos terminados en una muñeca. Su unidad de control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno. Normalmente se usa para realizar una tarea de manera cíclica, pudiéndose adaptar a otra sin cambios permanentes en su material “ (Martínez, 2012, pág. 3).

- **Federación Internacional de Robótica (IFR):** “Robot industrial de manipulación es una máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que pueden posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento” (Martínez, 2012, pág. 3).

Luego de leer las definiciones anteriores se concluye. Que un robot es una máquina desarrollada para el cumplimiento de diferentes tareas, sean en la industria o la diversión. La figura 23 muestra una máquina que consta de componentes mecánicos y electrónicos que funcionan sincronizada mente bajo la dirección de un sistema de control que realizar movimientos y actividades rutinarias del ser humano.

Figura 22. Arquitectura de un robot industrial



Fuente: (Martínez, 2012, pág. 82)

4.3. Robot scara

El robot scara diseñado en 1982 cuando el profesor Makino, un investigador japonés observó que el 80% de los movimientos de ensamblaje industrial era vertical y se

encuentran dentro del alcance horizontal del brazo humano. En base a esa observación planteo el diseño de un dispositivo robótico con un número limitado de grados de libertad 3 o 4, económico versátil para ser utilizado en labores de ensamble en líneas de producción automatizadas, la flexibilidad de este dispositivo que actualmente se utiliza en otra tareas que se desarrollan en sistemas de manufactura de productos de consumo masivo. El termino scara proviene de la palabra inglés "Selective compliance assemble robot arm" que se puede traducir como "brazo de cumplimiento selectivo para robot de montaje" (Checa, Luna, & Mosquera, 2009, pág. 1).

4.3.1. Estructura mecánica.

Generalmente los robots scara tienen dos articulaciones de revolución q_1 y q_2 , que permiten la rotación de los eslabones E1 y E2 en ángulos que van entre:

Ecuación 10. *Ángulos de rotación de eslabones*

$$-\frac{\pi}{2} \leq q_1 \leq \frac{\pi}{2}$$

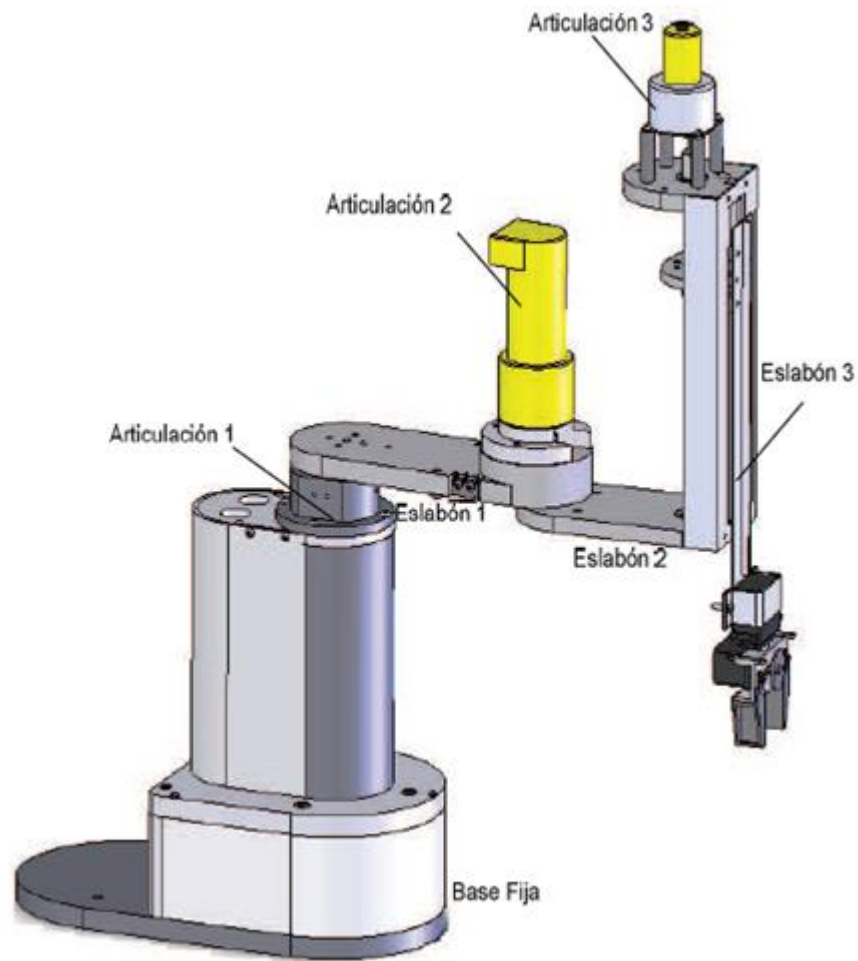
y

$$-\frac{\pi}{2} \leq q_2 \leq \frac{\pi}{2}$$

Fuente: (Ollero, 2001, pág. 76)

Se mueven respectivamente y una prismática en un plano perpendicular al del movimiento de los dos anteriores, proporcionar el desplazamiento del eslabón E3 (Ollero, 2001, pág. 76). En la figura 24 se observa el diseño de un robot scara, con tres grados de libertad con sus articulaciones y sus eslabones.

Figura 23. Robot scara 3 grados de libertad



Fuentes: (Boada & Morales, 2010, pág. 9)

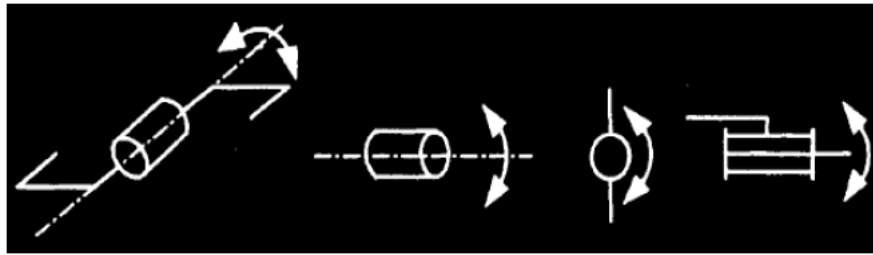
4.3.1.1. *Articulaciones.*

Las articulaciones permiten la unión entre dos eslabones. Las articulaciones definen el número de grado de libertad de un robot. En el robot scara se utilizan dos tipos:

➤ **Revoluta o rotación:**

Produce un movimiento de rotación sobre un eje común entre dos eslabones, la posición relativa entre los eslabones es expresada por el ángulo que forman los ejes. Esta articulación es la más utilizada y proporciona un grado de libertad. La figura 25 muestra la simbología utilizada para describir a una articulación de rotación.

Figura 24. Simbología de la articulación de rotación

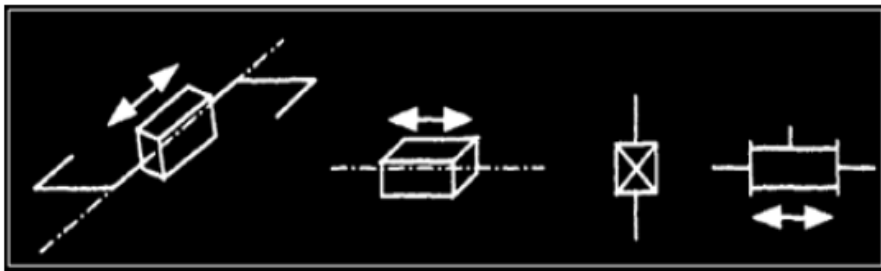


Fuentes: (Iñigo & Vidal, 2002, pág. 76)

➤ **Prismática:**

Permite un movimiento de traslación a lo largo del eje de la articulación. Esta articulación proporciona un grado de libertad. La figura 26 muestra la simbología utilizada para describir a una articulación de prismática.

Figura 25. Simbología articulación prismática



Fuentes: (Iñigo & Vidal, 2002, pág. 77)

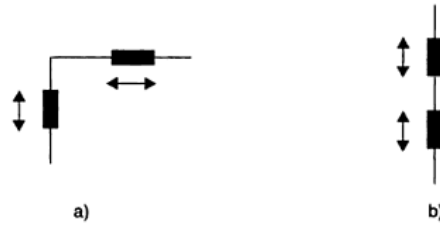
4.3.1.2. Eslabón.

Es la parte rígida que permite la unión entre dos articulaciones, generalmente constituyen el cuerpo del robot, por lo cual también determina su longitud.

4.3.1.3. Grados de libertad.

Es el número de movimientos independientes que puede realizar una articulación, el número de grados de libertad suele coincidir con el número de eslabones de la cadena cinemática, pero esto no siempre puede suceder como se observa en la figura 27a, una estructura con dos eslabones y dos articulaciones prismáticas que tiene 2 grados de libertad. Sin embargo, en la figura 27b se aprecia que existen dos eslabones y dos articulaciones pero solo tiene un grado de libertad.

Figura 26. Ejemplo de grados de libertad

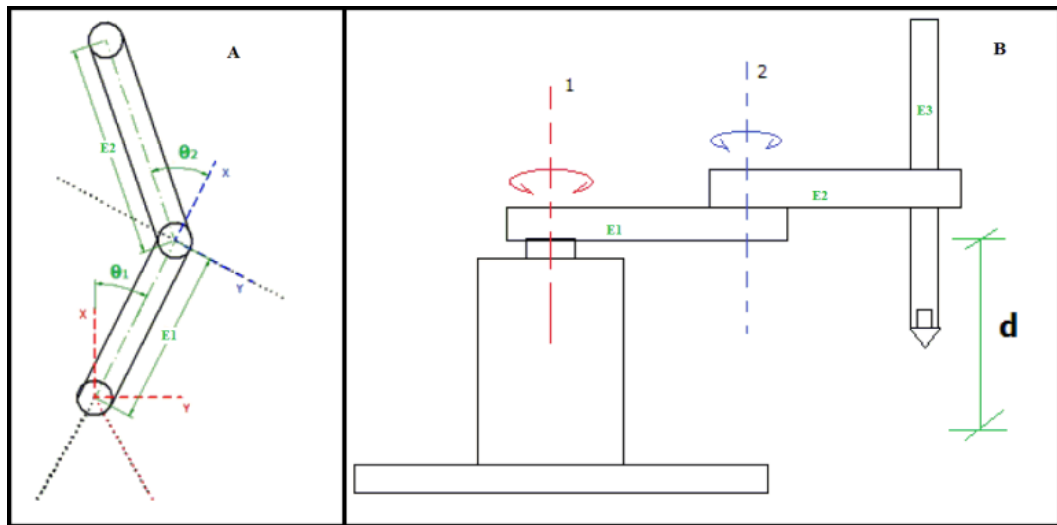


Fuente: (Ollero, 2001, pág. 86)

4.3.2. Dimensiones del robot scara.

El scara es un sistema mecánico considerado como cadena cinemática abierta, en la figura 28 se observa, los dos eslabones horizontales E1 y E2, mismos están unidos por las articulaciones rotativas de ejes verticales son los que van a determinar la longitud del robot y su espacio de trabajo. El eslabón E3 que realiza el movimiento vertical de traslación determina la altura del robot.

Figura 27. Dimensiones y parámetros del robot scara



Elaborado por: Edgar Changoluisa.

La tabla 2 describe los valores de las variables de la figura 28, que se utilizaron en la construcción del prototipo de robot scara, mismos que fueron determinados por el tamaño del tablero de damas que se utiliza.

Tabla 2. Dimensiones robot scara

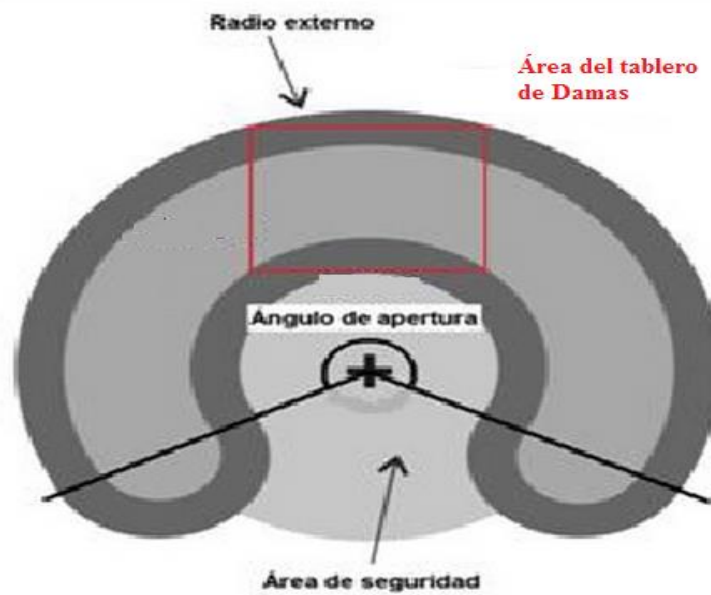
| Articulación | Tipo de Articulación | Variables | Angulo de giro/ desplazamiento | Longitud del eslabón |
|--------------|----------------------|------------|-----------------------------------|-------------------------|
| 1 | Revolución | Θ_1 | entre -90° y 90° | 29 cm |
| 2 | Revolución | Θ_2 | entre -90° y 90° | 29 cm |
| 3 | Prismática | d3 | entre 0 cm a 10 cm | 10 cm |

Elaborado por: Edgar Changoluisa.

4.3.3. Espacio de tarea.

También llamada área de trabajo, es la superficie en la cual el robot va realizar los movimientos, en la cual vamos a obtener datos como localización, posición. Generalmente expresadas en coordenadas cartesianas.

Figura 28. Área de trabajo robot scara



Elaborado por: Edgar Changoluisa

El robot scara, que se construyó para este proyecto, tiene tres grados de libertad, por lo cual su espacio de trabajo está determinado por la longitud y ángulos de rotación de los eslabones E1 y E3. En la figura 28, se muestran los movimientos que se van a

realizar, en dos cuadrantes del plano cartesiano porque las articulaciones de rotación, solo giran en un ángulo de 0° a 180°.

4.3.4. Actuadores.

Tienen la función de generar la fuerza motriz y transmitirla a los eslabones del robot, según las ordenes enviada por el sistema de control. Están ampliamente disponible en el mercado, son tipo neumático, hidráulico y electromagnéticos, además pueden ser modificados y adaptados de acuerdo a la aplicación del robot.

Para este proyecto, se han empleado tres actuadores electromagnéticos porque son de bajo costo, tienen precisión y una relativa facilidad de control.

4.3.5. Actuadores electromagnéticos

Son los que más se utilizan para la construcción de robots industriales, debido a su sencillez, control, variedad de tamaño y precisión, se clasifican en motores eléctricos de corriente alterna (AC) y los motores de corriente continua (DC) por ejemplo: motores de paso, servomotores.

En el proyecto se utilizó servomotores para transmitir movimiento a los eslabones, porque tienen incorporado una caja de reducción, un sistema de posicionamiento, y además son más livianos que los motores de paso.

4.3.5.1. Servomotores.

Es un dispositivo similar a un motor de corriente continua, que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición. Está conformado por un motor, una caja reductora y un circuito de control. Los servos se utilizan frecuentemente en sistemas de robótica y de radio control. Es posible modificar un servomotor para obtener un motor de corriente continua, esto se lo conoce como trincar al servo, si bien ya no tiene la capacidad de control del servo, conserva la fuerza, velocidad y baja inercia que caracteriza a estos dispositivos (TecMikro, 2014, pág. 1).

Partes principales:

- **Motor de corriente continua:** Es el elemento que le brinda movilidad al servo. Cuando se aplica un potencial a sus dos terminales, este motor gira en un sentido a su velocidad máxima.
- **Engranajes reductores:** se encargan de convertir gran parte de la velocidad de giro del motor de corriente continua en torque.
- **Circuito de control:** Este circuito es el encargado del control de la posición del motor. Recibe los pulsos de entrada y ubica al motor en su nueva posición dependiendo de los pulsos recibidos.

Los servomotores tienen 3 terminales:

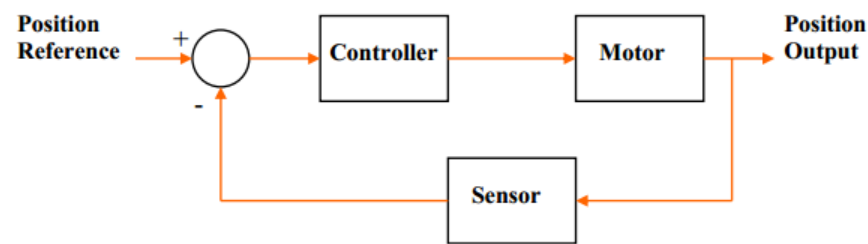
- **Terminal positivo:** recibe la alimentación del motor (4 a 8 voltios).
- **Terminal negativo:** referencia tierra del motor (0 voltios).
- **Entrada de señal:** recibe la señal de control del motor.

Los colores del cable de cada terminal varían con cada fabricante: el cable del terminal positivo siempre es rojo; el del terminal negativo puede ser marrón o negro; el del terminal de entrada de señal suele ser de color blanco, naranja o amarillo.

Control de posición:

Un sistema de control básico de un servo se describe en la figura 30. Los servos se controlan por medio de un pulso de ancho variable y frecuencia constante. El terminal de control se utiliza para el ingreso de este pulso. De acuerdo a las restricciones de rotación del servo motor, la posición neutral se define como aquella en la que el servomotor dispone del mismo ángulo de giro hacia la izquierda y hacia la derecha, es decir es una posición central a partir de la cual el eje puede girar el mismo ángulo en ambos sentidos (TecMikro, 2014, pág. 1).

Figura 29. Sistema básico de control de posición

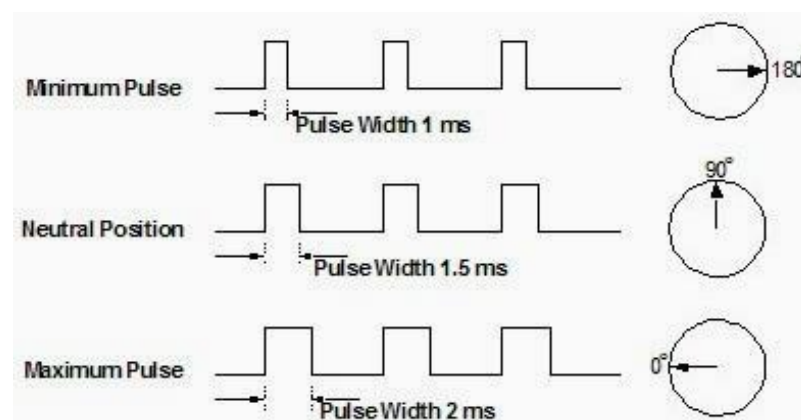


Fuente: (Control Systems Principles, 2009)

Cuando el motor servomotor recibe un pulso específico se mueve a la posición correspondiente y permanece en ella. Si se aplica una fuerza externa el servo ofrece resistencia a salir de dicha posición. La máxima fuerza que el motor servo puede ejercer está relacionada con uno de sus parámetros de mayor importancia: el torque máximo. Los servos no mantendrán su posición para siempre; se debe repetir el pulso para que el servo motor permanezca en dicha posición (TecMikro, 2014).

En la figura 31 se observa que cuando el servomotor recibe un pulso menor que 1,5 ms gira hasta una nueva posición un determinado ángulo hacia la derecha a partir de su posición neutral. Cuando el pulso es mayor que 1,5 ms ocurre lo contrario. Los límites mínimo y máximo de duración del pulso dependen de cada servomotor. Diferentes marcas, e inclusive diferentes modelos de la misma marca, normalmente tienen distintos límites mínimo y máximo. De manera general el pulso mínimo está alrededor de 1 ms y el máximo aproximadamente 2 ms (TecMikro, 2014, pág. 1).

Figura 30. Pulsos para control de un servomotor



Fuente: (TecMikro, 2014, pág. 1)

4.3.6. Cinemática inversa.

En el brazo scara tenemos que resolver los dos ángulos de las articulaciones de rotación. En este tipo de brazos el eje Z no interviene en la cinemática inversa porque es un resultado en sí mismo. Para el cálculo sólo debemos tener en cuenta los ejes X e Y, y la longitud de las articulaciones (Martínez, 2012, pág. 65).

Para calcular la cinemática inversa de un sistema con dos grados de libertad es importante determinar que conocemos y cuáles son las variables que existen.

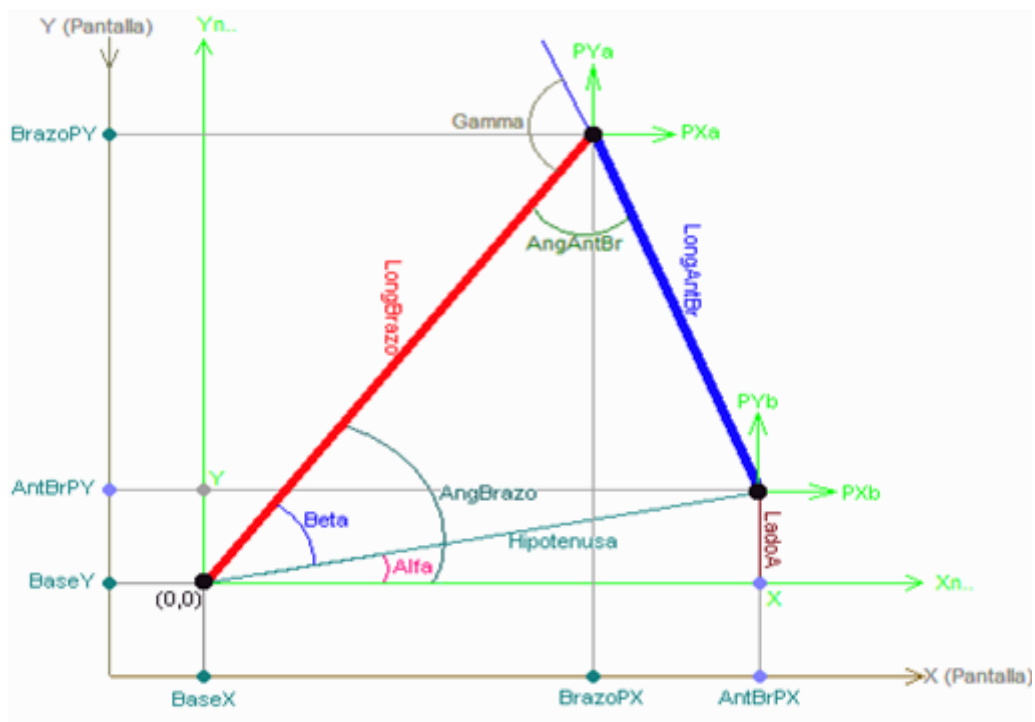
Conocemos:

- Longitud de los eslabones del brazo E1 y antebrazo E2.
- Ubicación final en coordenadas x,y.

Variables:

- Angulo del brazo.
- Angulo antebrazo.

Figura 31. Problema de la cinemática inversa de un robot scara



Elaborado por: Edgar Changoluisa.

En la figura 32 el triángulo formado por LongBrazo, LongAntBr e Hipotenusa suele ser del tipo irregular, sobre todo si los dos catetos LongBrazo y LongAntBr no son iguales. Para resolver los ángulos de este tipo de triángulo hay que aplicar el Teorema del Coseno.

Teorema del Coseno

En un triángulo el cuadrado de cada lado es igual a la suma de los cuadrados de los otros dos menos el doble producto del producto de ambos por el coseno del ángulo que forman (Aritor, 2008):

Ecuación 11. *Teorema del coseno*

$$a^2 = b^2 + c^2 - 2bc \cos \alpha$$

Fuente: (Aritor, 2008)

En la ecuación 12 se detalla que en las fórmulas trigonométricas se usa la función arcocoseno "Acos()" y un tipo de arcotangente llamado "Atan2()". No es lo mismo Atan(Y/X) que Atan2(Y, X) Se diferencia la una de la otra, además de la sintaxis, en que Atan2(), dependiendo de los signos contenidos en los valores X e Y tiene presente el cuadrante en el que se encuentra y puede recorrer los cuatro cuadrantes. Por contrapartida la función Atan() sólo tiene efecto en los cuadrantes 1 y 4 (Proyectos Robóticos, 2008).

Ecuación 12. *Formulas de la resolución geométrica de la cinemática inversa*

```
Hipotenusa = sqrt( (X^2)+(Y^2) )

Alfa =      Atan2(Y, X)

Beta =      Acos( (LongBrazo^2 - LongAntBr^2 + Hipotenusa^2) / (2 * LongBrazo * Hipotenusa) )

AngBrazo = Alfa + Beta

Gamma =     Acos ( (LongBrazo^2 + LongAntBr^2 - Hipotenusa^2) / (2 * LongBrazo * LongAntBr) )

AngAntBr = Gamma - 180
```

Fuente: (Proyectos Robóticos, 2008)

CAPÍTULO 5

ANÁLISIS Y DISEÑO DEL SISTEMA

5.1. Definición de requisitos

Como se ha mencionado en el primer capítulo, en el alcance del proyecto, este consta de tres módulos que son: visión artificial, inteligencia artificial y robótica, a continuación se detalla los requerimientos funcionales y requerimientos no funcionales para la implementación de cada uno de ellos.

5.1.1. Requerimientos funcionales.

➤ Módulo de inteligencia artificial

El módulo permite que un ser humano juegue damas contra un robot. Por medio de un algoritmo de búsqueda conocido como poda alfa y beta mencionado en el ítem 3.4.1.2 del capítulo 3, se evalúa y selecciona la mejor jugada para ganar el juego en el menor tiempo posible. En la interfaz de Usuario permite configurar el nivel de dificultad del juego, el tiempo máximo de cada movimiento y elegir el color de fichas con las que juega el contrincante.

➤ Módulo de robótica

Por medio de un prototipo de brazo tipo scara se moviliza las fichas dentro del tablero de damas, utilizando la ecuación 12 para resolver el cálculo de cinemática inversa, se encontró la mejor ruta para mover una ficha desde el punto A hacia un punto B. El prototipo no va a manipular las ficha, para movilizarlas va a utilizar un electroimán que le permite mover las fichas con rapidez y precisión.

➤ Módulo de visión artificial

Con una cámara web se captura imágenes de los movimientos del contrincante y el módulo debe identificar que ficha se movió, en tiempo real. El sistema debe tener una interfaz para poder modificar los parámetros de los filtros y algoritmos utilizados para el reconocimiento.

➤ Módulo de Interfaz de usuario

El sistema debe contar con una interfaz gráfica para el usuario que va permitir configurar los parámetros de los módulos de inteligencia y visión artificial.

➤ **Módulo de adquisición de datos**

El módulo debe permitir el envío y recepción de datos entre el robot y el software de control de una forma sencilla.

5.1.2. Requisitos no funcionales.

Debido a los factores externos y la complejidad que se presenta, al momento de predecir todos los posibles escenarios que se deben analizar se definió que es indispensable construir un prototipo el cual va a permitir trabajar en un entorno controlado para las respectivas pruebas.

En el prototipo va a existir una estructura de metal en la cual se va a colocar el dispositivo de captura de imágenes, sistema de iluminación, tarjeta electrónica de control, tablero de juego de damas y el robot scara.

El tablero de damas tiene 32 fichas las cuales el contrincante va a mover para realizar las jugadas. El módulo de visión artificial tiene como objetivo principal capturar, procesar e identificar cuáles fueron los movimientos que se realizaron en el tablero. Para capturar las imágenes se va a utilizar una cámara web. El tablero y las fichas deben ser diseñados especialmente para este proyecto, hay que tomar en cuenta elegir colores que permitan la fácil segmentación de los objetos en la imagen capturada.

Una buena iluminación es una parte primordial en el escenario porque gracias a ella se pueden obtener mejores resultados en el reconocimiento de imágenes, no se debe tener una iluminación muy clara porque produce cambios en el contraste de los colores y por lo tanto puede afectar el procesamiento de las imágenes, es importante tener una iluminación natural la que no provoque sombras, contraste altos ni bajos.

5.2. Diseño del sistema

En la siguiente sección se describirá los componentes del sistema, detallado las características generales, especificaciones técnicas y funcionales.

5.2.1. Diseño de hardware.

5.2.1.1. Diseño mecánico.

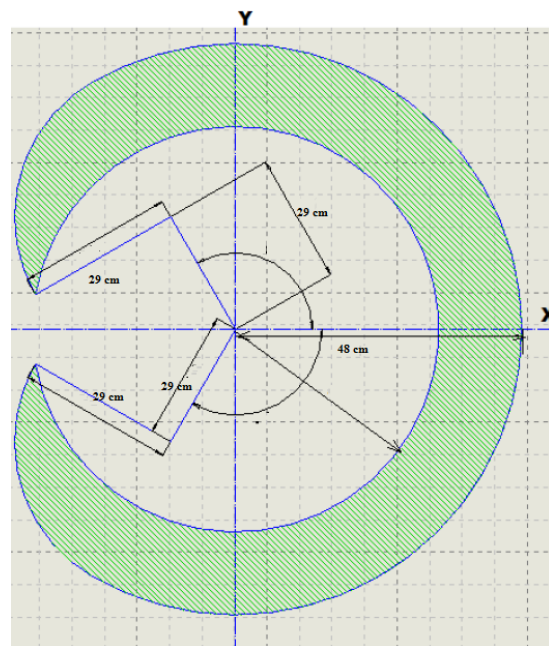
En la tabla 3 se han propuesto las dimensiones de acuerdo a equipos comerciales destinados a similares aplicaciones.

Tabla 3. *Características técnicas*

| Descripción | Valores |
|-------------------------|---------|
| Alcance Máximo | 58 cm |
| Desplazamiento vertical | 10cm |
| Carga movilizar | 0,01 kg |
| Posición de montaje | Base |
| Grados de libertad | 3 |

Elaborado por: Edgar Changoluisa

Figura 32. Ejes y campos de rotación del robot scara



Elaborado por: Edgar Changoluisa

El prototipo del robot scara consta de dos articulaciones rotacionales y una prismática por lo cual tiene 3 grados de libertad.

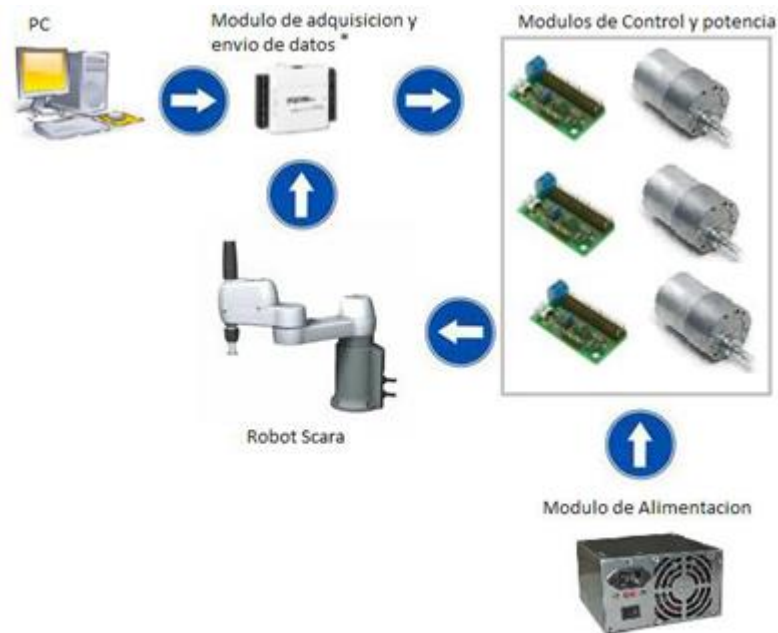
El valor de la carga que se va a movilizar, está determinado por la masa de la ficha que se usa en nuestro tablero de damas.

En la figura 35 se observa que cada eslabón tiene un valor de 29 centímetros por lo tanto el alcance máximo del brazo robótico es de 58cm.

5.2.1.2. Sistema electrónico.

En lo que se refiere a la parte electrónica en la figura 34 se describe los 4 módulos que se usan para el control del brazo robótico.

Figura 33. Módulos propuesto para el control



Fuente: (Guzmán & Sevillano, 2012, pág. 8)

- **Módulo de Alimentación:** La Fuente de poder que alimentará todo el prototipo, será escogida tomando en cuenta la carga total que representan los tres motores que moverán el robot, las tarjetas electrónicas compuestas por un microcontrolador y los elementos de potencia para manejar los motores.
- **Módulo de adquisición y envío de datos:** La función de este módulo es encargarse de la transmisión de las señales de control a los módulos de potencia desde la PC.

- **Módulo de control y potencia:** la función de este es la de recibir e interpretar las señales de control y convertir esta señal en una respuesta de movimiento de cada motor.

5.2.1.3. Sistema de control.

El prototipo de robot scara tiene tres grados de libertad, el movimiento de cada eslabón puede referirse al sistema de coordenadas del eslabón anterior. Por lo tanto, es posible describir la trayectoria deseada en función del movimiento de cada una de las articulaciones. El prototipo debe mover una ficha desde el punto inicial A hacia un punto final B. Este movimiento puede ser ejecutado en infinitas trayectorias. Para el cálculo de la trayectoria se utiliza de la cinemática inversa geométrica.

Debido a que los dos eslabones del prototipo tienen el mismo tamaño, el modelo de cinemática inversa que vamos a usar es el geométrico el cual se lo describe en el capítulo 4 página 54.

El flujo de la figura 35 describe los pasos que se realizan durante el proceso de controlar el prototipo de brazo robótico scara.

- **Ingreso de coordenadas del punto:** Las posiciones en que las fichas se ubican dentro del tablero de damas son fijas, por lo tanto cuando el robot debe realizar un movimiento se envía el valor de las coordenadas en X y Y de la casilla donde está la ficha o va a estar.
- **Realizar cálculos de ángulos y desplazamiento:** Por medio de la ecuación 12 de cinemática inversa, se obtiene el valor de los ángulos para cada una de las articulaciones rotacionales. El desplazamiento de la articulación prismática es fijo con un valor de 10 cm, que representa la distancia aproximada entre el electroimán del prototipo y el tablero de damas.
- **Envío de ángulos:** Porque todos los cálculos se realizan en la PC, por medio de un puerto USB se va a enviar el valor de los ángulos a la tarjeta de control.
- **Controlador:** El módulo de control luego de recibir el valor de los ángulos, procede a enviar el comando para que cada servomotor (articulación), se ubique en la respectiva posición.

Figura 34. Diagrama de flujo del modelo de cinemática inversa



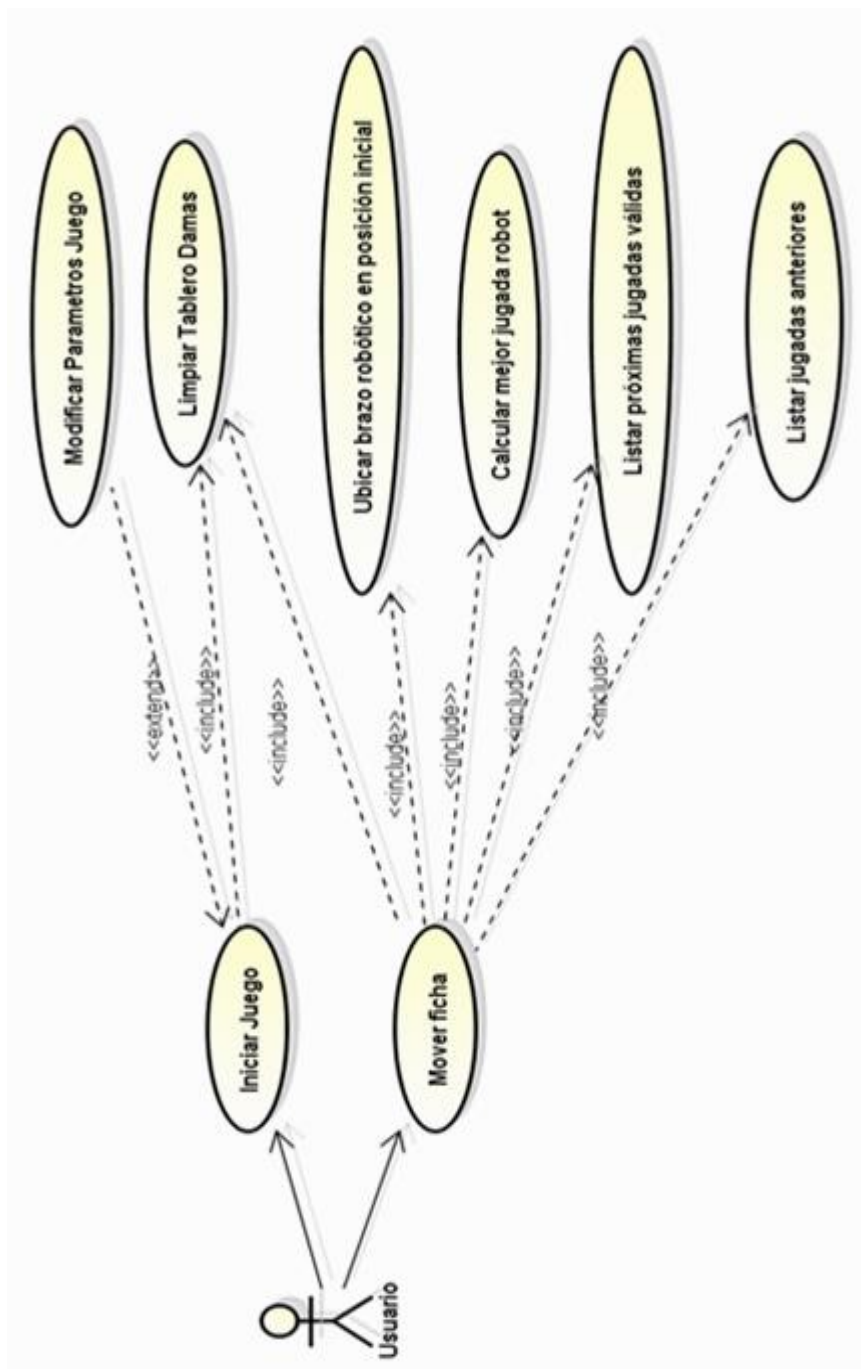
Elaborado por: Edgar Changoluisa

5.2.2. Diseño software.

5.2.2.1. Diagrama de casos de uso.

Para comprender de una mejor manera el sistema en la figura 36 de modela los casos de uso del mismo, esto ayudará a identificar los actores y acciones y su manera de interactuar entre los mismos. Luego de revisar los requisitos del proyecto se han contemplado los siguientes casos de uso:

Figura 35. Casos de uso



Elaborado por: Edgar Changoluisa.

En las tablas 4, 5, 6 se detalla el diagrama de caso de uso.

Tabla 4. *Actores del sistema*

| Actor | Descripción |
|---------|--|
| Usuario | Es el humano que jugará damas contra el robot. |

Elaborado por: Edgar Changoluisa

Tabla 5. *Especificación del caso de uso: Iniciar juego*

| | |
|-------------------------|---|
| Caso de uso | Iniciar Juego |
| Propósito | Empezar el juego |
| Actores | Usuario |
| Precondiciones | Las fichas deben estar colocadas dentro del tablero para iniciar un nuevo juego. El brazo del robot debe estar en posición inicial. |
| Post condiciones | Todos los componentes de sistema estarán listos para iniciar la partida |

Elaborado por: Edgar Changoluisa

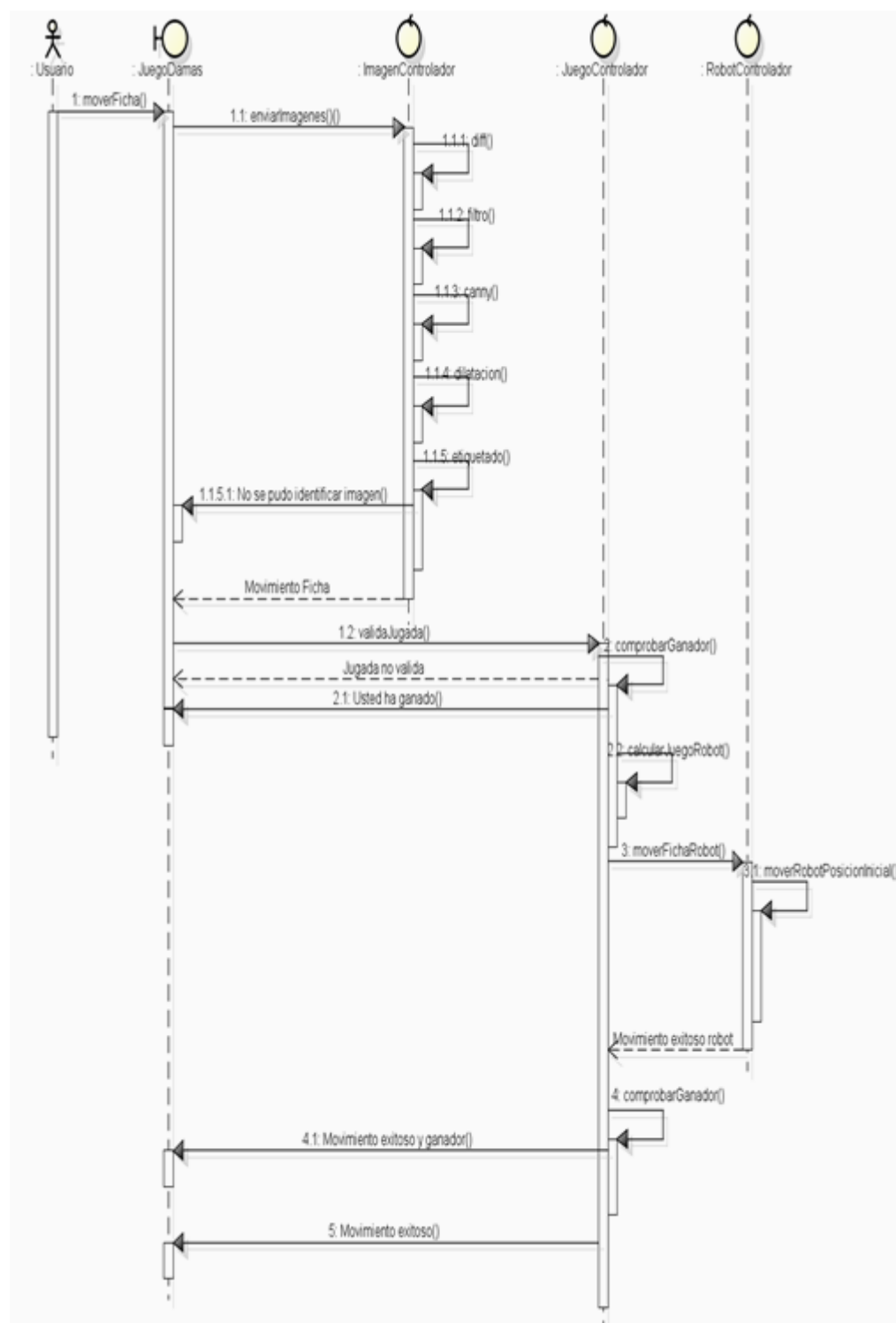
Tabla 6. *Especificación del caso de uso: Mover ficha*

| | | |
|------------------------|---|---|
| Propósito | El actor indica que ha movido una ficha y espera que el robot realice su movimiento para continuar con el juego | |
| Actores | Usuario | |
| Precondiciones | El humano debe haber realizado una jugada válida. | |
| Pos condiciones | El robot moverá la ficha y espera que el jugador humano realice un jugada dentro del tablero. | |
| Flujo | Pasos | Acción |
| Principal | 1 | El usuario presiona el botón que indica su jugada |
| | 2 | El kernel ejecuta el algoritmo de identificación de la jugada |
| | 3 | El kernel evalúa la mejor jugada y envía la señal al brazo para que ejecute el movimiento |
| | 4 | El brazo robótico se mueve hacia la ficha que será movida. |
| | 5 | El electroimán del brazo robótico baja y toma la ficha a mover |
| | 6 | El brazo robótico se mueve hacia la posición de destino de la ficha |
| | 7 | El electroimán suelta la ficha |
| | 8 | El brazo robótico vuelve a la posición cero y espera la jugada del humano. |

Elaborado por: Edgar Changoluisa

5.2.2.2. Diagrama de secuencia.

Figura 36. Diagrama de secuencia del caso de uso mover ficha



Elaborado por: Edgar Changoluisa

La figura 37 contiene la información de la implementación del escenario, incluyendo la secuencia cronológica de los objetos, clases y los mensajes intercambiados entre los objetos del caso de uso Mover Ficha, se utilizó este caso de uso porque es el más usado durante todo el juego.

En la tabla 7 se detallan todos los pasos del diagrama de secuencia de la figura 37. La tabla contiene el número del paso, el mensaje o estímulo que se realiza, la interfaz u operación que participa, la descripción de las acciones que se realizan y la respuesta que se retorna.

Tabla 7. *Descripción de los pasos del diagrama de secuencia*

| # Paso | Mensaje/ Estímulo | Interface/ Operación | Descripción | Respuesta |
|--------|-------------------|-------------------------------|--|----------------------|
| 1.0 | moverFicha | Usuario- JuegoDamas | El sistema recibe un mensaje que el usuario movió una o varias fichas. | ir al siguiente paso |
| 1.1 | enviarImagenes | JuegoDamas- ImagenControlador | Por medio de la cámara web se captura y se envía la imagen del tablero de damas. | ir al siguiente paso |
| 1.1.1 | diff | ImagenControlador | Se realiza una comparación entre la imagen anterior y nueva, para determinar cuáles son los cambios. | ir al siguiente paso |
| 1.1.2 | filtro | ImagenControlador | Se aplican filtros de gauss y mediana, para eliminar el ruido de la imagen que se está procesando | ir al siguiente paso |
| 1.1.3 | canny | ImagenControlador | Buscamos los bordes de las fichas en la imagen que estamos procesando | ir al siguiente paso |
| 1.1.4 | dilatacion | ImagenControlador | Dilatamos de la imagen, con el fin de resaltar los bordes de las fichas. | ir al siguiente paso |
| 1.1.5 | etiquetado | ImagenControlador | Se procede a identificar las fichas que fichas se movieron y su ubicación actual en el tablero | ir al siguiente paso |

Continúa...

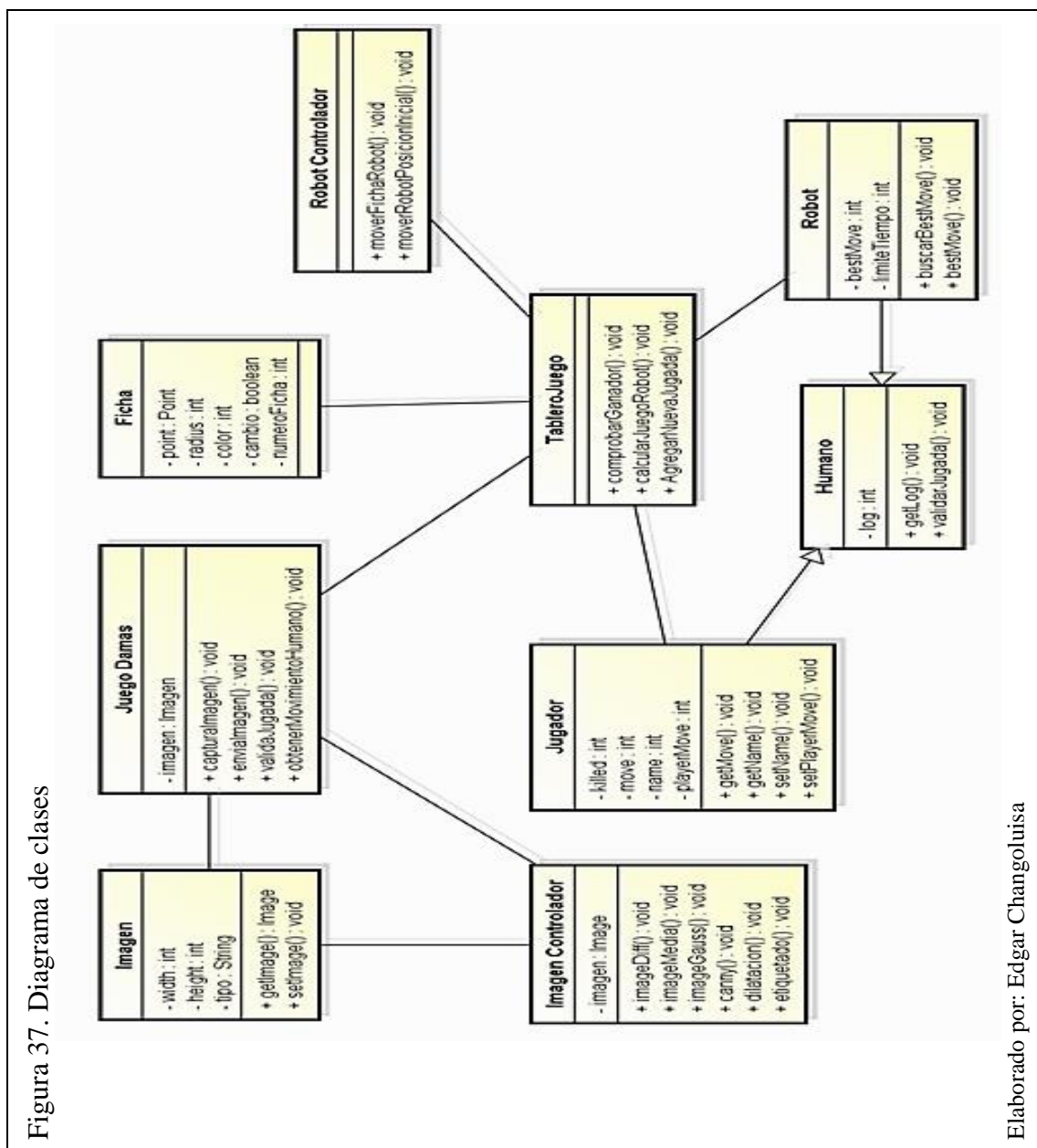
| # Paso | Mensaje/ Estímulo | Interface/ Operación | Descripción | Respuesta |
|--------|-------------------------------|-----------------------------------|--|--|
| 11.5.1 | No se pudo identificar imagen | ImagenControlador-JuegoDamas | Valida si la imagen capturada se pudo procesar correctamente. | Si, ir al siguiente paso, caso contrario finaliza todo el proceso. |
| 1.2 | validaJugada | JuegoDamas-JuegoControlador | Valida si la jugada realizada por el usuario, está permitida de acuerdo a las reglas del juego de Damas del Anexo 1. | Si, ir al siguiente paso, caso contrario finaliza todo el proceso y envía el mensaje "Jugada no Valida". |
| 2.0 | comprobarGanador | JuegoControlador | Verifica si con el movimiento realizado el usuario, gana el Juego. | Si, va al paso 2.1, caso contrario va al paso 2.2 |
| 2.1 | Usted ha ganado | JuegoControlador | Finaliza el juego | envía el mensaje "Usuario a ganado" |
| 2.2 | calculaJuegoRobot | JuegoControlador | Ejecuta los algoritmos de búsqueda para obtener la mejor jugada que el robot puede realizar. | ir al siguiente paso |
| 3.0 | moverFichaRobot | JuegoControlador-RobotControlador | Realiza los cálculos necesarios, para obtener el valor de los ángulos de las articulaciones del robot scara. | ir al siguiente paso |
| 3.1 | moverRobotPosicionInicial | RobotControlador | Se envía los valores de los ángulos, a la tarjeta de control, para que se ejecuten los respectivos movimientos. | Movimiento éxito |
| 4.0 | comprobarGanador | RobotControlador | Comprueba si el robot gana el juego con el movimiento realizado. | Si, va al paso 4.1, caso contrario va al paso 5.0 |

Continúa...

| # Paso | Mensaje/ Estímulo | Interface/ Operación | Descripción | Respuesta |
|--------|------------------------------|---------------------------------|---|----------------------|
| 4.1 | Movimiento exitoso y ganador | JuegoControlador- JuegoDamas | Finaliza el juego. | "Robot ha ganado" |
| 5.0 | Movimiento exitoso | JuegoControlador- JuegoDamas | Termina el proceso y se reinicia el flujo al paso 1 | "Movimiento exitoso" |

Elaborado por: Edgar Changoluisa.

5.2.2.3. Diagrama de clases.



En la figura 38 se muestran las clases principales más representativas del sistema, estas se han extraído al identificar los casos de uso.

En las tablas 8 a la 16 se realiza la descripción, en detalle de las clases pertenecientes del sistema. Cada una de ellas tiene la descripción de sus atributos y métodos al igual que las relaciones entre ellos.

Tabla 8. *Detalle del diagrama de clase, clase Image*

Clase: Imagen

Paquete: com.robotic.vision.artificial.conversion

Detalle: Contiene los datos de las cualidades de las imágenes del sistema.

Conexiones

| Conector | Multiplicidad | Fuente | Destino |
|------------|---------------|--------|--------------------|
| Asociación | Varios a uno | Imagen | Juego Damas |
| Asociación | Varios a uno | Imagen | Imagen Controlador |

Atributos

| Nombre | Visibilidad | Tipo | Notas | Valor Inicial |
|--------|-------------|--------|-------------------------------|---------------|
| width | Private | int | Indica el ancho de la imagen. | 400 |
| height | Private | int | Indica el alto de la imagen. | 400 |
| Tipo | Private | String | Indica el tipo de imagen. | Null |

Operaciones

| Nombre | Visibilidad | Alcance | Retorna | Parámetro |
|-------------|-------------|------------------------------------|---------|-----------|
| getImage() | Public | Devuelve el objeto de tipo Imagen. | Imagen | ninguno |
| setImagen() | Public | Indica el alto de la imagen. | Void | Imagen |

Elaborado por: Edgar Changoluisa

Tabla 9. *Detalle del diagrama de clase, clase Imagen Controlador*

Clase: Imagen Controlador

Paquete: com.robotic.vision.artificial.algoritmos

Detalle: Contiene las operaciones que se realizan a las imágenes en el sistema.

Conexiones

| Conector | Multiplicidad | Fuente | Destino |
|------------|---------------|--------------------|-------------|
| Asociación | Uno a Varios | Imagen Controlador | Imagen |
| Asociación | Varios a uno | Imagen Controlador | Juego Damas |

Atributos

| Nombre | Visibilidad | Tipo | Notas | Valor Inicial |
|--------|-------------|-------|-------------------------|---------------|
| imagen | Private | Image | Atributo de tipo Image. | Null |

Operaciones

| Nombre | Visibilidad | Alcance | Retorna | Parámetros |
|--------------|-------------|---|---------|------------|
| imageDiff() | Public | Realiza la diferencia entre dos imágenes. | void | ninguno |
| imageMedia() | Public | Aplica el filtro de la mediana en una imagen para eliminar ruido. | void | ninguno |
| imageGauss() | Public | Aplica el filtro de Gauss en una imagen para eliminar ruido. | void | Ninguno |
| canny() | Public | Busca los borde de las figuras. | void | Ninguno |
| Nombre | Visibilidad | Alcance | Retorna | Parámetros |
| dilatación() | Public | Dilata los bordes de las figuras. | void | Ninguno |
| etiquetado() | Public | Reconoce y clasifica las figuras de la imagen que se procesa. | void | Ninguno |

Elaborado por: Edgar Changoluisa

Tabla 10. *Detalle del diagrama de clase, clase Juego Damas*

Clase: Juego Damas

Paquete: com.robotic.vision.artificial.gui

Detalle: Contiene la interfaz gráfica del sistema. Permite al usuario interactuar con el robot durante el juego de damas.

Conexiones

| Conector | Multiplicidad | Fuente | Destino |
|------------|---------------|-------------|--------------------|
| Asociación | Uno a Varios | Juego Damas | Imagen |
| Asociación | Uno a Varios | Juego Damas | TableroJuego |
| Asociación | Uno a Varios | Juego Damas | Imagen Controlador |

Atributos

| Nombre | Visibilidad | Tipo | Notas | Valor Inicial |
|--------|-------------|-------|------------------------|---------------|
| imagen | Private | Image | Atributo de tipo Image | Null |

Operaciones

| Nombre | Visibilidad | Alcance | Retorna | Parámetros |
|----------------------------|-------------|--|---------|------------|
| capturaImagen() | Public | Captura la Imagen del tablero de damas para ser procesado. | void | ninguno |
| enviaImagen() | Public | Envía la Imagen para ser procesada. | void | ninguno |
| validaJugada() | Public | Valida si la jugada realizada por el usuario es válida, según las reglas del Anexo 1 | void | Ninguno |
| obtenerMovimiento Humano() | Public | Dibuja en la interfaz del usuario, las siguientes jugadas que el usuario puede realizar. | void | Ninguno |

Elaborado por: Edgar Changoluisa

Tabla 11. *Detalle del diagrama de clase, clase Jugador*

Clase: Jugador

Paquete: com.robotic.inteligencia.artificial.gui

Detalle: Clase que contiene los estados de los usuarios.

Conexiones

| Conector | Multiplicidad | Fuente | Destino |
|------------|---------------|--------------------|---------------|
| Asociación | Uno a Uno | Jugador | TableroJuegos |
| Herencia | Uno a Uno | Imagen Controlador | Humano |

Atributos

| Nombre | Visibilidad | Tipo | Notas | Valor Inicial |
|------------|-------------|--------|---|---------------|
| killed | Private | int | Indica si el Usuario tiene una ficha muerta. | 0 |
| Move | Private | int | Indica el que movimiento realizo | 0 |
| Name | Private | String | Nombre del participante. La PC tiene el nombre de "Robot" | Participante |
| playerMove | Private | int | Indica si debe realizar la nueva jugada. | 0 |

Operaciones

| Nombre | Visibilidad | Alcance | Retorna | Parámetros |
|-----------------|-------------|---|---------|------------|
| getMove() | Public | Obtiene el valor del movimiento que realizo el jugador. | int | ninguno |
| getName() | Public | Obtiene el Nombre del Jugador | String | ninguno |
| setName() | Public | Envía el nombre del Jugador | void | String |
| setPlayerMove() | Public | Envía si debe realizar la siguiente Jugada. | void | Int |

Elaborado por: Edgar Changoluisa

Tabla 12. *Detalle del diagrama de clase, clase Ficha*

Clase: Ficha

Paquete: com.robotic.datos.matrices

Detalle: Clase que contiene las cualidades que tiene una ficha durante el juego.

Conexiones

| Conector | Multiplicidad | Fuente | Destinó |
|------------|---------------|--------|--------------|
| Asociación | Varios a Uno | Ficha | TableroJuego |

Atributos

| Nombre | Visibilidad | Tipo | Notas | Valor Inicial |
|-------------|-------------|---------|--|---------------|
| point | Private | ponit | Indica cuales son las coordenadas en x,y de la ficha. | (0,0) |
| Radius | Private | int | Indica el radio de la ficha. | 0 |
| Color | Private | int | Existen 3 colores de fichas, 1: roja, 2: azul, 0: blanca o vacío | 0 |
| Cambio | Private | boolean | Indica si la ficha fue movida por el usuario. | False |
| numeroFicha | Private | int | Indica que ficha es, de las 32 fichas que existen en el tablero. | 1 |

Elaborado por: Edgar Changoluisa

Tabla 13. *Detalle del diagrama de clase, clase TableroJuego*

Clase: TableroJuego

Paquete: com.robotic.datos.matrices

Detalle: Clase controla las jugadas que realizan los jugadores.

Conexiones

| Conector | Multiplicidad | Fuente | Destino |
|------------|---------------|--------------|-------------------|
| Asociación | Uno a uno | TableroJuego | Robot Controlador |
| Asociación | Uno a varios | TableroJuego | Ficha |
| Asociación | Varios a uno | TableroJuego | Juego Damas |
| Asociación | Uno a uno | TableroJuego | Jugador |
| Asociación | Uno a uno | TableroJuego | Robot |

Operaciones

| Nombre | Visibilidad | Alcance | Retorna | Parámetros |
|----------------------|-------------|---|---------|------------|
| comprobarGanador() | Public | Comprueba si existe en ganador. | void | ninguno |
| calcularJuegoRobot() | Public | Realiza el cálculo de la mejor jugada para el robot. | void | ninguno |
| agregarNuevaJugada() | Public | Agrega las nuevas jugadas al tablero de damas, para que la interfaz gráfica, la dibuje. | void | Ninguno |

Elaborado por: Edgar Changoluisa

Tabla 14. *Detalle del diagrama de clase, clase Humano*

Clase: Humano

Paquete: com.robotic.inteligencia.artificial.gui

Detalle: Clase que contiene los movimientos del usuario humano.

Conexiones

| Conector | Multiplicidad | Fuente | Destino |
|----------|---------------|--------|--------------------|
| Herencia | Uno a varios | Humano | Imagen Controlador |

Atributos

| Nombre | Visibilidad | Tipo | Notas | Valor Inicial |
|--------|-------------|--------|--|---------------|
| log | Private | string | Contiene los log de las jugadas realizadas por el usuario. | Null |

Operaciones

| Nombre | Visibilidad | Alcance | Retorna | Parámetros |
|-----------------|-------------|--|---------|------------|
| getLog() | Public | Obtiene el log, de las jugadas realizadas por los usuarios e imprime en la interfaz. | void | ninguno |
| validarJugada() | Public | Válida la juga que realizo el usuario, si esta es válida de acuerdo al anexo 1. | void | ninguno |

Elaborado por: Edgar Changoluisa

Tabla 15. *Detalle del diagrama de clase, clase Robot*

Clase: Robot

Paquete: com.robotic.inteligencia.artificial.algoritmo

Detalle: Clase que realiza la implementación de la poda alfa & beta, para buscar y elegir la mejor jugada que puede realizar el robot.

Conexiones

| Conector | Multiplicidad | Fuente | Destino |
|------------|---------------|--------|--------------|
| Asociación | Uno a uno | Robot | TableroJuego |

Atributos

| Nombre | Visibilidad | Tipo | Notas | Valor Inicial |
|--------------|-------------|------|---|---------------|
| bestMove | Private | int | Obtiene el valor del movimiento, para poder hacer la poda. | 0 |
| limiteTiempo | Private | int | Limita el tiempo de al realizar la poda, crea el nivel de dificultad. | 1000 ms |

Operaciones

| Nombre | Visibilidad | Alcance | Retorna | Parámetros |
|-----------------|-------------|--|---------|------------|
| buscaBestMove() | Public | Por medio de la poda alfa y beta realiza la búsqueda y elección de la mejor jugada que va a realizar el robot. | void | ninguno |
| bestMove() | Public | envía la mejor jugada que el robot puede realizar. | move | ninguno |

Elaborado por: Edgar Changoluisa

Tabla 16. *Detalle del diagrama de clase, clase Robot Controlador*

Clase: Robot Controlador

Paquete: com.robotic.robot.scara

Detalle: Clase que realiza los cálculos del valor de los ángulos, para cada una de las 3 articulaciones y envía los valores para la tarjeta de control.

Conexiones

| Conector | Multiplicidad | Fuente | Destino |
|------------|---------------|-------------------|--------------|
| Asociación | Uno a uno | Robot Controlador | TableroJuego |

Operaciones

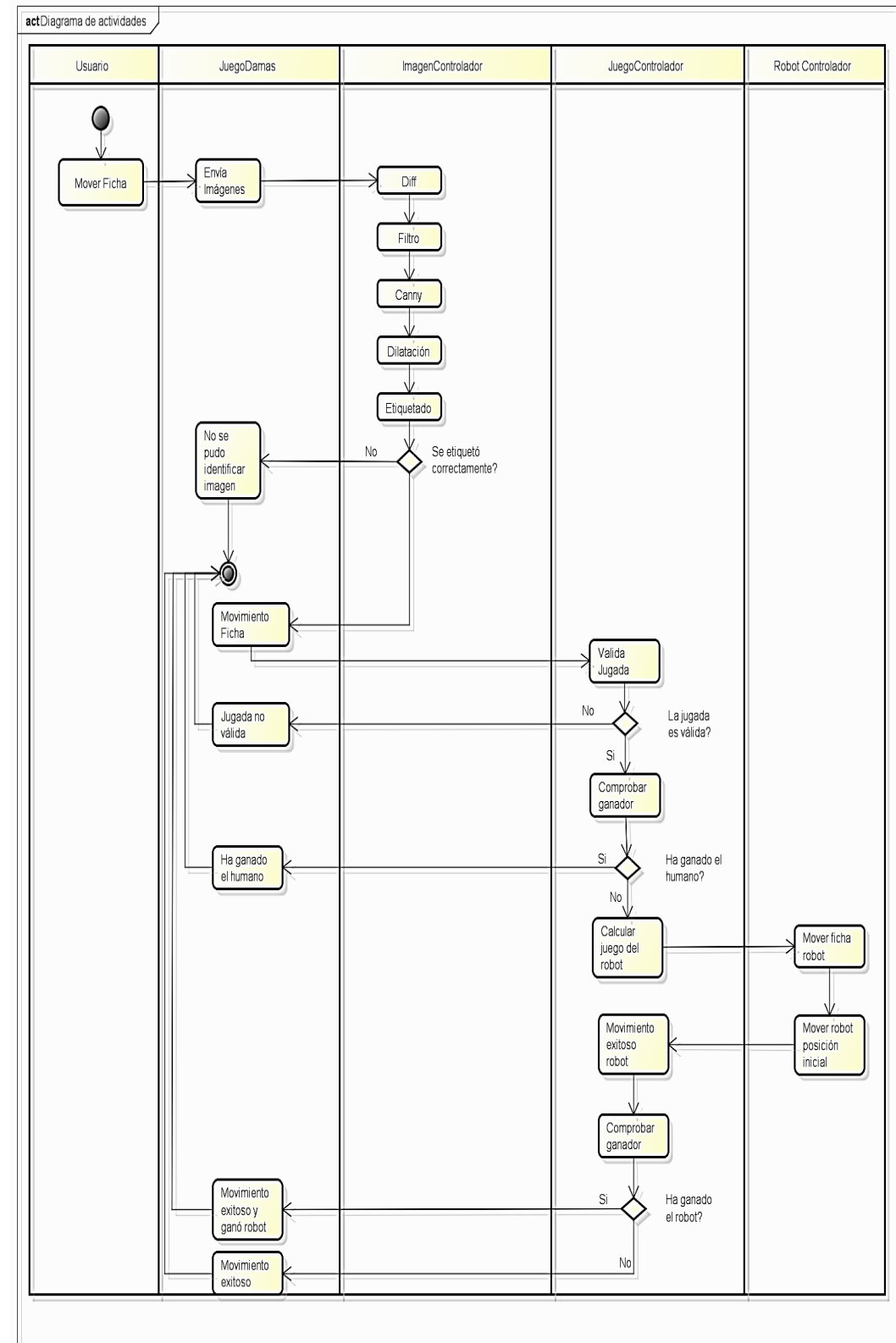
| Nombre | Visibilidad | Alcance | Retorna | Parámetros |
|-------------------------------|-------------|---|---------|------------|
| moverFichaRobot() | Public | Realiza el cálculo del valor de los ángulos, para cada una de las articulaciones, del nuevo movimiento. | void | ninguno |
| moverRobot Posicion Inicial() | Public | Envía el valor de los ángulos del nuevo movimiento a la tarjeta de control. | void | ninguno |

Elaborado por: Edgar Changoluisa

5.2.2.4. *Diagrama de actividades.*

En la figura 39 se describe cuáles son las actividades que realizan los actores y cuales el sistema, se obtiene a partir del diagrama de secuencia.

Figura 38. Diagrama de actividades

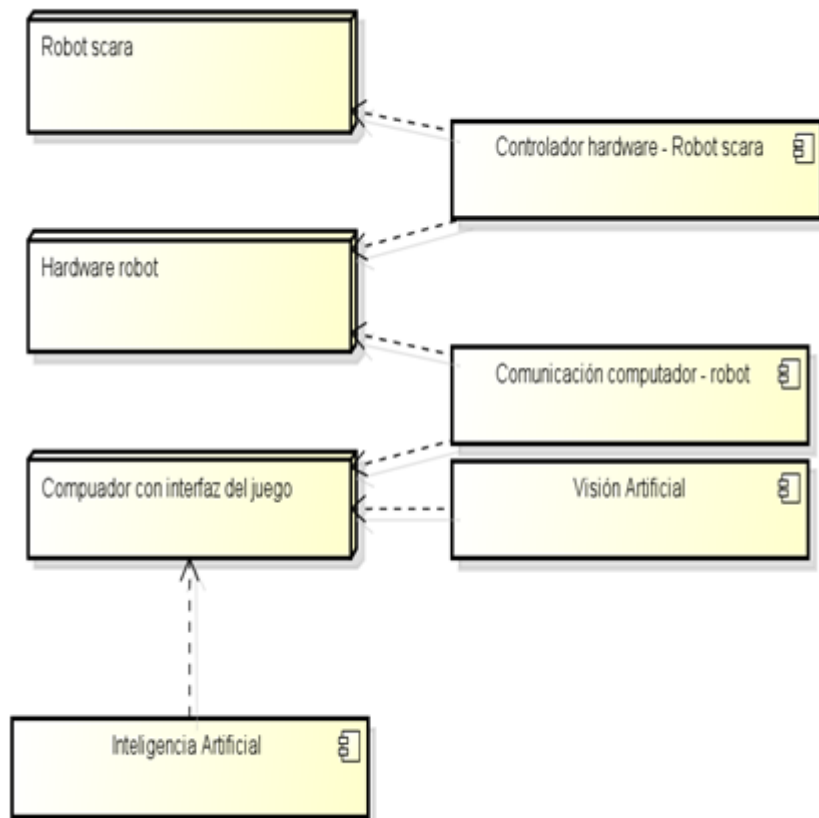


Elaborado por: Edgar Changoluisa.

5.2.2.5. Diagrama de despliegue.

En la figura 40 se modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. (Sparx Systems, 2007, pág. 1)

Figura 39. Diagrama de despliegue



Elaborado por: Edgar Changoluisa.

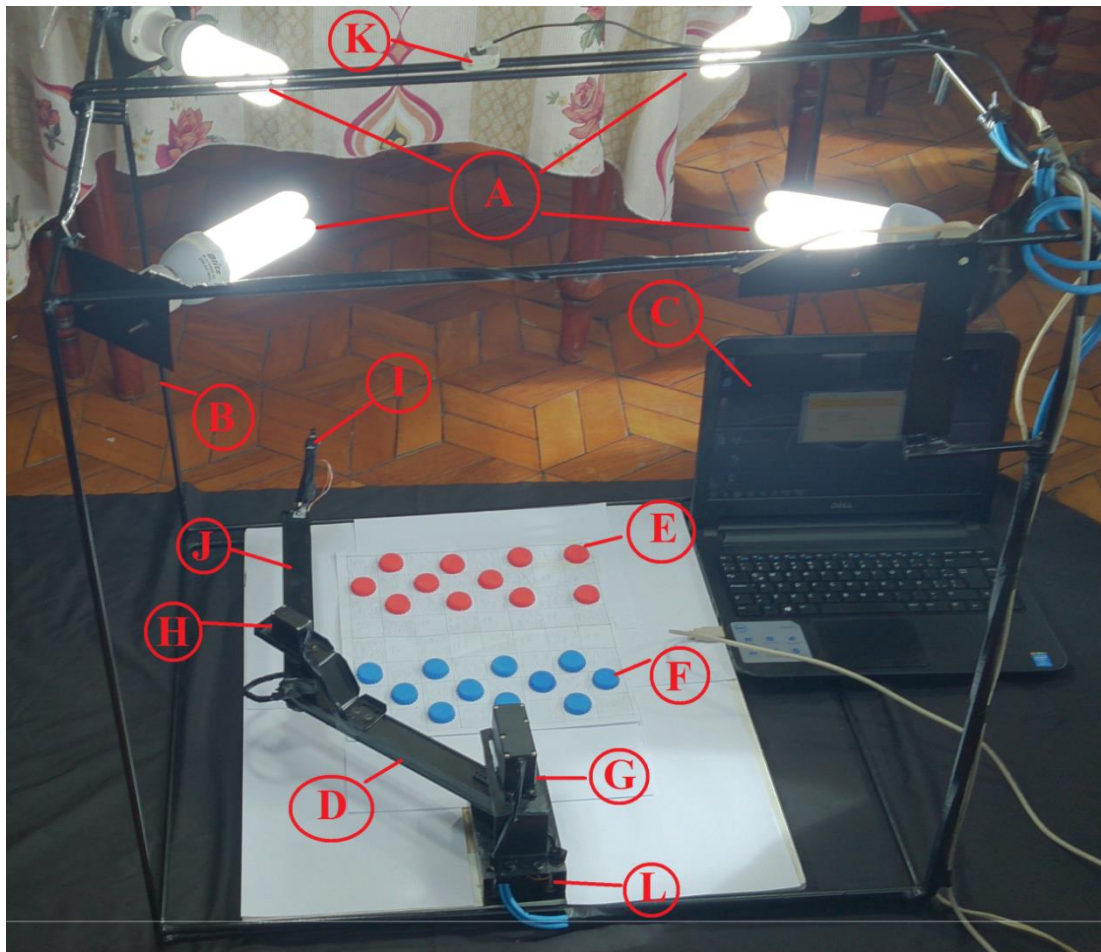
CAPÍTULO 6

CONSTRUCCIÓN DEL SISTEMA

6.1. Entorno de trabajo

Para el correcto funcionamiento del sistema se debe tomar en cuenta los diferentes factores externos que intervienen, mismos que pueden alterar la información y modificar el funcionamiento del sistema, por lo cual se construyó un prototipo que tiene las siguientes dimensiones 60x60x70 m³. Para poder realizar las pruebas en un ambiente controlado, en la figura 41 se puede observar el prototipo y sus componentes.

Figura 40. Prototipo y sus componentes



Elaborado por: Edgar Changoluisa.

A continuación se describen los elementos del prototipo que se construyó, para el proyecto, mismo que se observan en la figura 41.

- A. Sistema de iluminación:** De acuerdo a lo revisado en el capítulo 2, el sistema de iluminación, es un elemento de un sistema de visión artificial, es importante para eliminar ruidos y evitar colocar filtros al momento de procesar las imágenes. En el proyecto se utiliza un sistema de iluminación frontal, la descripción del mismo está en el ítem 2.3.2.4 del capítulo antes citado. Existen varias Fuentes de Luz, el detalle de este tema se encuentra en el ítem 2.3.3. Para el proyecto se utilizaron focos fluorescente porque minimizan los reflejos y además porque la luz que emiten es blanca.
- B. Estructura del Prototipo:** Para colocar los elementos del sistema de visión artificial, el robot scara y el tablero de damas, se construyó una estructura compuesta de varilla de hierro, de 60cm de ancho, 60 cm de alto y 70 cm de largo.
- C. Computadora- Interfaz de usuario:** Se utiliza una laptop, para ejecutar el programa que contiene la interfaz de usuario y la lógica para jugar damas, el ordenador debe tener instalado la versión de java 1.7 como mínimo, el sistema operativo es Windows 7, mínimo debe tener 4 gigas en RAM debido a que los programas hechos en java consume más memoria RAM que procesador cuando se ejecutan.
- D. Eslabón del Hombro:** Es la parte rígida que permite la unión entre la articulación del hombro con la del brazo. En la tabla 2, del capítulo 4 de robótica, se describe la longitud del eslabón del hombro, que es de 29 cm.
- E. Ficha primer jugador:** En el juego de damas se utiliza por lo general fichas de color negro y blanco, para nuestro proyecto se han cambiado estos colores, el primer jugador va a utilizar las fichas de color rojo. El cambio de los colores de las fichas se hizo para poder segmentar por color, de una manera más fácil porque los colores van estar en rangos muy alejados. Se utilizó acuarelas para pintar las fichas, para obtener una reflexión especular (ver ítem 2.3.1.1 del capítulo 2). Para que el electroimán del robot pueda atrapar a la ficha, utilizamos como ficha una tapa de gaseosa, debido a su forma, tamaño y el material del cual está construida.

- F. Ficha segundo jugador:** se va a utilizar una ficha de color azul para el segundo jugador, con las mismas consideraciones descritas para la ficha del primer jugador.
- G. Servomotor del hombro:** En el capítulo 4 en el ítem 2.3.5.1 se describen la cualidades de los servomotores, de las cuales se resalta que son muy fáciles de utilizar porque tiene un circuito integrado. En la tabla 18 m1 hace referencia a las características del servomotor que usamos en la articulación del hombro del robot.
- H. Servomotor del brazo:** En la tabla 19 m2 hace referencia a las características del servomotor que se usa en la articulación del brazo del robot, en los cálculos realizados en el sitio web (robotshop, 2014), para obtener el valor del torque necesario para cada articulación se determina que el torque mínimo es de 4kg para esta articulación.
- I. Electroimán:** En el alcance del proyecto, se detalla que nuestro prototipo no es un manipulador y para mover la ficha va a usar un electroimán. Se utiliza la bobina de un relé de 5v para el electroimán del prototipo.
- J. Eslabón del brazo:** Es la parte rígida que permite la unión entre la articulación del brazo con la de la mano (electroimán). En la tabla 2, del capítulo 4 de Robótica, se describe la longitud del brazo, que es de 29 cm.
- K. Cámara web:** Como se detalla en el ítem 2.4.1 del capítulo 2, para la captura y envío de imágenes a una pc se utiliza una cámara web. En este proyecto se utiliza un cámara web Genius LOOK 312P que tiene una resolución de 352 x 288 pixel.
- L. Base de robot:** Esta pieza fija el robot al área de trabajo por medio de pernos.

6.2 Implementación de los módulos

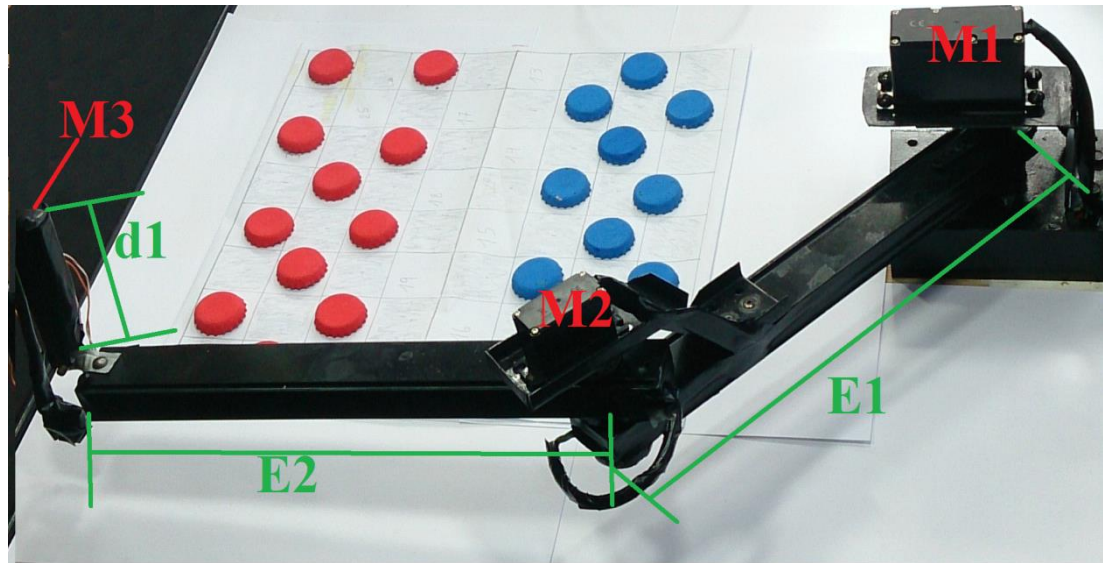
El sistema se encuentra diseñado por módulos independientes que interactúan entre sí mediante mensajes de entrega y salida, esto permite que el sistema se pueda adaptar a cualquier cambio o mejoramiento.

6.2.1 Módulo de robótica.

6.2.1.1. Diseño mecánico.

En la figura 42 se observar los elementos mecánicos del prototipo del robot scara, las dimensiones fueron determinadas por el tamaño del tablero de damas que tiene un tamaño de 28x28 cm.

Figura 41. Prototipo robot scara



Elaborado por: Edgar Changoluisa

Tabla 17. Dimensiones de los elementos del prototipo del robot scara

| Motor | Tipo de Articulación | Variables | Angulo de giro/ desplazamiento | Longitud del eslabón |
|-------|----------------------|-----------|-----------------------------------|-------------------------|
| M1 | Revolución | E1 | entre -90° y 90° | 29 cm |
| M2 | Revolución | E2 | entre -90° y 90° | 29 cm |
| M3 | Prismática | D1 | entre 0 cm a 10 cm | 10 cm |

Elaborado por: Edgar Changoluisa

En la tabla 17 se detalla los valores de las variables, que aparecen en la figura 42.

- **M1:** Es la articulación rotatoria que pertenece al hombro del robot, como se detalla en el ítem 4.3.5.1 del capítulo 4, los servomotores por lo general, tienen un ángulo de trabajo entre -90 y 90 grados. La longitud del eslabón para esta articulación es de 29cm.

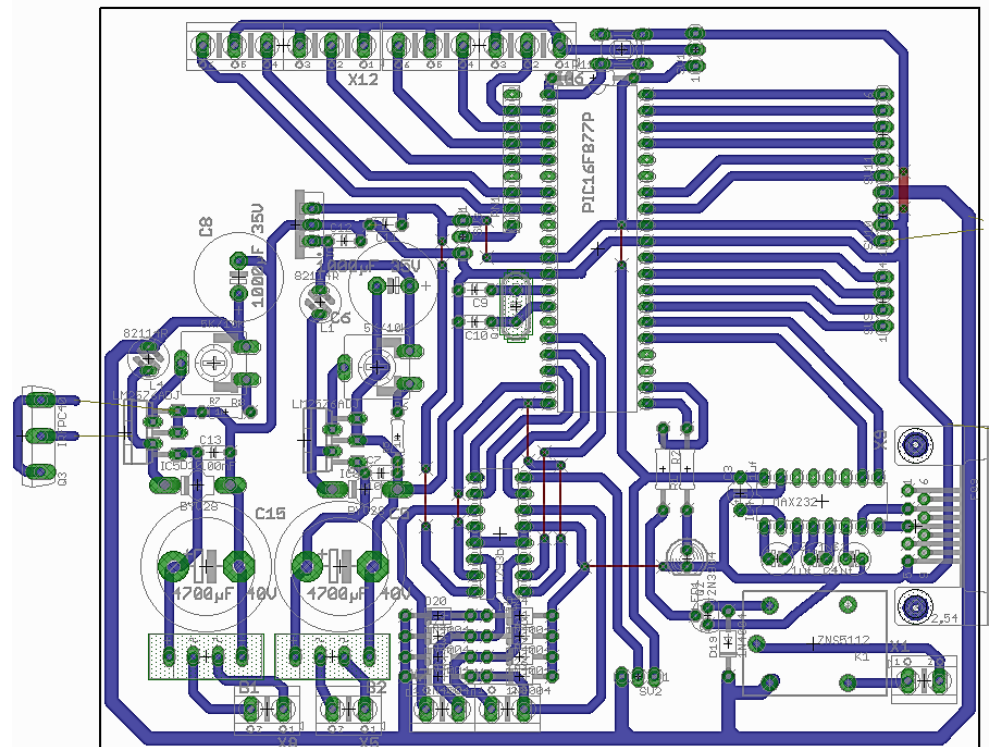
- **M2:** Es la articulación rotatoria que pertenece al brazo del robot, tiene un ángulo de trabajo entre -90 y 90 grados. La longitud del eslabón para esta articulación es de 29cm.
- **M3:** Es la articulación prismática que pertenece a la mano del robot, tiene un ángulo de trabajo entre -90 y 90 grados. El desplazamiento de esta articulación es de 10 cm, este valor se determinó, por la distancia que existe entre el tablero de damas y la mano del robot.

6.2.1.2. Sistema electrónico.

En la parte de diseño se mencionan las 4 partes principales que se debe considerar al momento de construir la tarjeta de control del prototipo de robot scara.

- **Módulo de Alimentación:** La construcción de una Fuente de 5 voltios y mínimo 2 amperios la misma que dotará de energía a los otros módulos de manera independiente, en la figura 43 se observa el diagrama esquemático de la Fuente de alimentación, sus elementos están detallados en el anexo 2.
- **Módulo de adquisición y envío de datos:** La comunicación entre la pc y el microcontrolador se la va a realizar mediante la interfaz estándar USB, en la figura 44 se observa, que el pin 25 se usa para la recepción de datos y 26 para la transmisión de datos.
- **Módulo de control y potencia:** En la figura 44 se muestra el diagrama esquemático del sistema de control, dicho módulo controla todos los eventos que el prototipo del robot scara puede realizar. Los servomotores contienen un circuito que separa el módulo de control del módulo de potencia.

Figura 44. Diagrama PCB de la tarjeta madre del sistema



Elaborado por: Edgar Changoluisa

En la figura 45 se muestran los mismos componentes de la figura 43 y 44, pero con el dimensionamiento exterior para la placa impresa, los agujeros de montaje necesarios, la correcta colocación y distribución de los elementos, pistas que serán impresas. En el anexo 2 se listan los componentes electrónicos del circuito.

Figura 45. Circuito de construcción de la tarjeta madre (vista frontal)



Elaborado por: Edgar Changoluisa

La figura 46 es el resultado del diseño y construcción del circuito principal de la fuente de alimentación y el circuito de la tarjeta de control del sistema, esta placa es nuestro hardware en una vista frontal, aquí sus elementos más importantes como el microcontrolador, condensadores, señales de RX y TX.

6.2.1.3. Servomotores.

En la tabla 18 describen las características básicas de los servomotores utilizados en la construcción del prototipo de robot scara.

Tabla 18. *Características de los servomotores usados en el prototipo*

| Motor# | Serie | Bearing Type | Speed | Torque kg./cm | Size in Millimeters | Weight grams |
|--------|--------------|-------------------|---|-----------------------------|---------------------------|--------------|
| M1 | HS-805BB M | Dual Ball Bearing | (4.8V/6.0V): 0.19 / 0.14 | (4.8V/6.0V): 19.8 / 24.7 | 65.79 x 29.97 x 57.40 | 153.09 |
| M2 | HS-5485HB | Top Ball Bearing | (4.8V/6.0V): 0.20 / 0.17 | (4.8V/6.0V): 5.2 / 6.4 | 39.88 x 19.81 x 37.85, | 45.08 |
| M3 | HS-81 | None | (4.8V/6.0V): 0.11 / 0.09 sec @ 60 deg | (4.8V/6.0V): 2.6 / 3.0 | 29.72 x 11.94 x 29.46 | 16.44 |

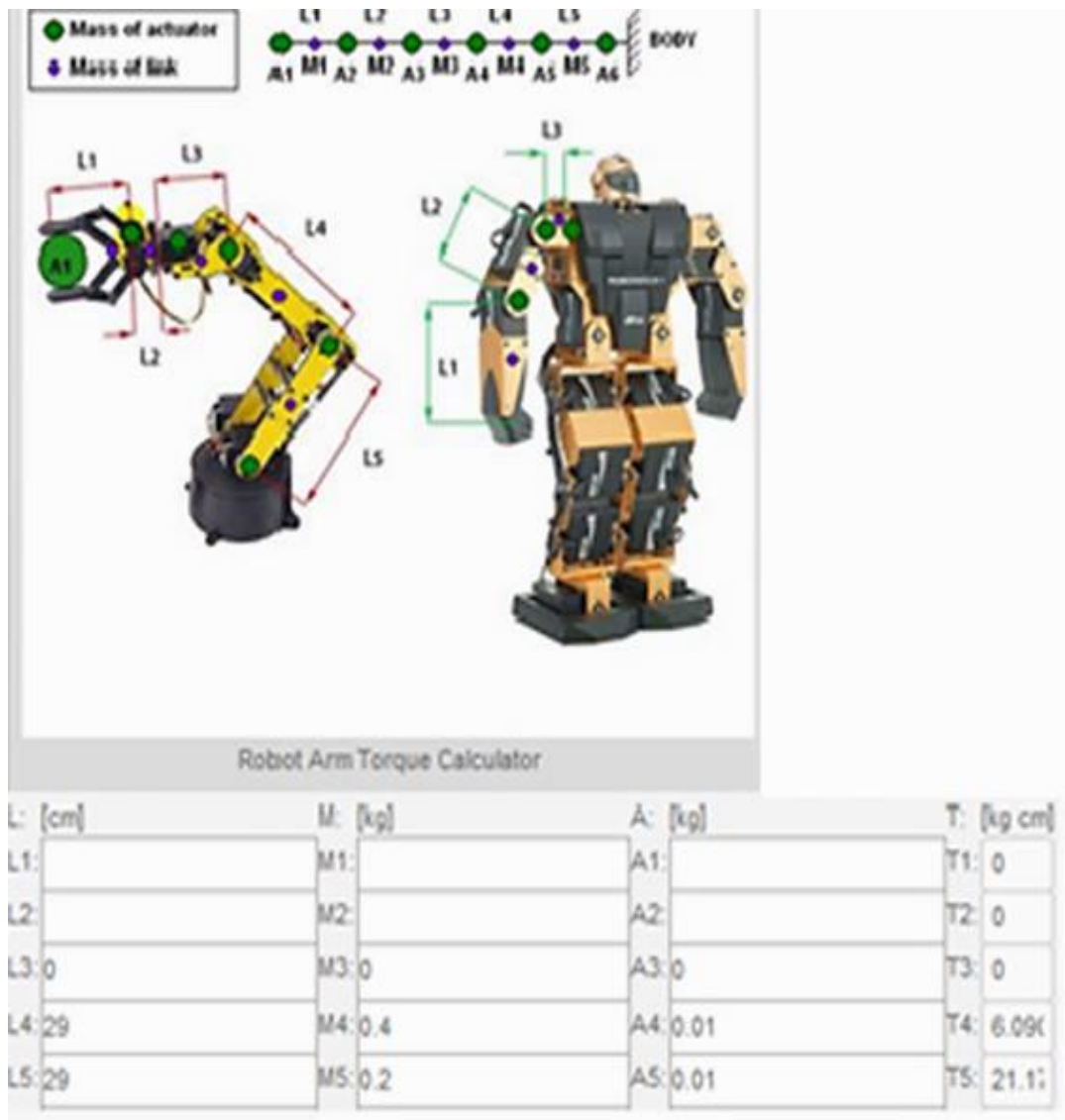
Elaborado por: Edgar Changoluisa

A continuación vamos hacer un análisis de la tabla 18 y sus valores.

- **Moto#:** Son las siglas que se utilizan en la figura 42 para indicar las articulaciones que tiene nuestro prototipo. M1 articulación hombro, M2 articulación del brazo, M3 articulación de la mano.
- **Serie:** Es el número único que da el fabricante a cada servomotor para poder identificarlo.
- **Bearinf type:** Indica que tipo de cojinetes que utiliza el servomotor. El cojinete o rodamiento es un elemento mecánico que reduce la fricción entre un eje y las piezas conectadas a éste por medio de rodadura, que le sirve de apoyo y facilita su desplazamiento.
- **Speed:** representa la velocidad que el servomotor puede alcanzar, en la tabla 18 se describen 2 velocidades, las cuales dependen del voltaje con que se alimenta el servomotor, puede ser 4,8V o 6,0V. La máxima velocidad es cuando se conecta a 6.0 V.

- **Torque:** Depende de la cantidad de voltaje con que se alimente al mismo, este alcanza su máximo torque cuando se le envía 6,0V. El torque es el momento de fuerza que ejerce un motor sobre el eje de transmisión de potencia.
- **Size:** se describe en milímetros las dimensiones que tiene el servomotor.
- **Weight:** indica la masa en gramos del servomotor.

Figura 46. Cálculo del torque para un robot



Fuente: (robotshop, 2014)

Para poder elegir correctamente el modelo de cada servomotor que se describe en la tabla 18, el cálculo del torque es importante para poder mover cada eslabón.

Al elegir un servo con torque menor al necesitado, no se puede movilizar los eslabones con precisión y existe el riesgo que los motores se quemen o se rompan los engranes. Por otro lado si utilizáramos servomotores con más torque del necesario, el costo de nuestro prototipo se incrementa.

El cálculo del torque para cada motor, se realizó en el sitio web (robotshop, 2014), en la figura 47 se detalla los parámetros solicitados para el cálculo del torque, que son L la longitud del eslabón, M la masa del eslabón y A la masa que se va a movilizar.

6.2.2. Programación del software.

6.2.2.1. Definición del algoritmo de búsqueda.

El algoritmo a utilizar para que el sistema elija la mejor jugada es la minmax con poda alfa y beta, una descripción más amplia de este algoritmo la encontramos en el ítem 3.4.1.2 del capítulo 3 del presente trabajo. La primera llamada a la función recursiva es:

MINIMAX_ALFABETA (posiciónactual, 0, JUGADOR-UNO, MAXINT, MININT)

El algoritmo minimax con poda alfa-beta es el siguiente:

```

MINIMAX_ALFABETA( posición, profundidad, jugador, umbralUSO, umbralPASO)
comienzo
  Si SUFICIENTE (posición, profundidad) entonces
    resultado.VALOR := ESTATICA (posición, jugador);
    resultado.CAMINO := NULO;
    return resultado;
  sino
    Sucesores := GENMOV (posición, jugador);
    Si EstaVacía (Sucesores) entonces
      resultado.VALOR := ESTATICA (posición, jugador);
      resultado.CAMINO := NULO;
      return resultado;
    sino
      por cada sucesor de Sucesores
        resultadoSucesor := MINIMAX_ALFABETA (sucesor, profundidad+1,
        CONTRARIO (jugador), -umbralPASO, -umbralUSO);
        // el mejor resultado se va guardando en umbralPASO
        Si umbralPASO < - resultadoSucesor.VALOR entonces
          umbralPASO := - resultadoSucesor.VALOR;
        resultadoMejor.CAMINO := sucesor + resultadoSucesor.CAMINO;
      fin si;
    // umbralUSO se usa para podar los hermanos

```

```

        Si umbralPASO >= umbralUSO entonces
            resultadoMejor.VALOR := umbralPASO;
            return resultadoMejor;
        finsi;
    fin por;
    resultadoMejor.VALOR := umbralPASO;
    return resultadoMejor;
    fin sino;
fin sino;
fin MINIMAX_ALFABETA;

```

A continuación se incluye el fragmento de código de la implementación del algoritmo de la poda alfa y beta en JAVA:

```

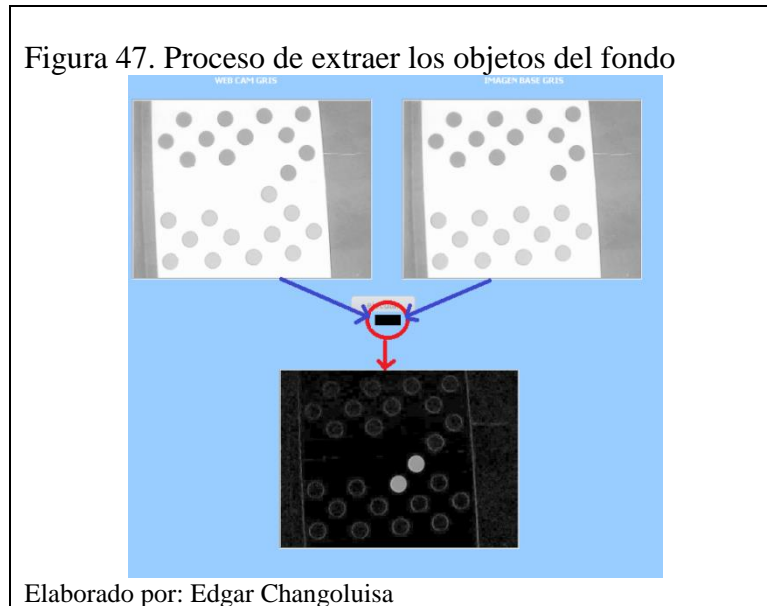
public int negaScout(Posiciones position, int depth, int alpha, int beta) {

    log.imprimir("Parametro negaScout  " + position.getLegalMoves() + " " + depth +
    " " + alpha + " " + beta,true);
    c++;
    //if(depth == 0 || message.isHaltPressed() || (new Date()).getTime() -
    moveTime.getTime() > (long)timeLimit)
    if (depth == 0 || (new Date()).getTime() - moveTime.getTime() > (long) timeLimit) {
        return evaluation(position);
    }
    int best = -10000;
    int tmpBeta = beta;
    ArrayList allLegalMoves = position.getLegalMoves();
    int numberOfLegalMove = allLegalMoves.size();
    for (int currentMove = 0; currentMove < numberOfLegalMove && best < beta;
    currentMove++) {
        int value = -negaScout(position.makeMove((Movimientos)
    allLegalMoves.get(currentMove)), depth - 1, -tmpBeta, -Math.max(alpha, best));
        if (value > best) {
            if (tmpBeta == beta || depth <= 2) {
                best = value;
                if (depth == moveLimit) {
                    bestMove = currentMove;
                }
            } else {
                best = -negaScout(position.makeMove((Movimientos)
    allLegalMoves.get(currentMove)), depth - 1, -beta, -value);
                if (depth == moveLimit) {
                    bestMove = currentMove;
                }
            }
        }
        tmpBeta = Math.max(alpha, best) + 1;
    }
    log.imprimir(" movimientos bestes " + best,true);
    return best;
}

```


6.2.3. Procesamiento y análisis de imágenes.

Para detectar los objetos que aparecen en la imagen se ha usado la técnica de segmentación basada en movimiento. Para utilizar esta técnica es importante tener un fondo estático y objetos en movimiento. Esta técnica se basa sencillamente en restar a la imagen del fondo con los objetos la imagen del fondo sin los objetos, como resultado tenemos una imagen solo con los objetos, como se observa en la figura 48.



Como paso previo al procesamiento de la imagen, se le debe capturar la imagen desde la web cam, para lo cual utilizamos el siguiente código:

```
public void runWebCam(int retardo){  
    log.imprimir("CamaraIniciada....",true);  
    if(cap.isOpened()){  
        try {  
            Thread.sleep(retardo);  
            cap.read(getImagenWebCam());  
            if(!imagenWebCam.empty()){  
                oK=true;  
            }  
            else{  
                oK=false;  
            }  
        } catch (Exception ex) {  
            oK=false;  
            onWebCam=false;  
            Logger.getLogger(CapturarImagenWebCam.class.getName()).log(Level.SEVERE, null, ex);  
        }  
    }  
}
```

La librería que se va a utilizar para el procesamiento de imágenes es OpenCV, la cual tiene implementadas las funciones necesarias para realizar todos los pasos descritos en la figura 37 del diagrama de secuencia que los realiza la clase ImageControlador. Se incluye un fragmento del código de cómo se realiza la invocación a la función de OpenCV:

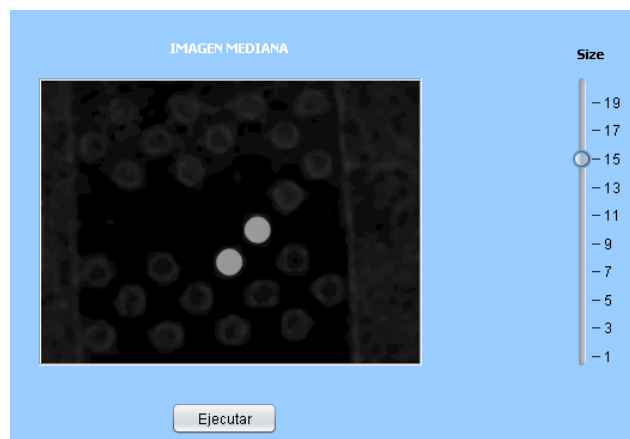
```
public void imageToImageGris(Mat imageBase, Mat imageNueva) {
    setImgendiff(new Mat(imageBase.width(), imageBase.height(), imageBase.type()));
    Core.absdiff(imageBase, imageNueva, imagenDiff);
}
```

6.2.4. Aplicación de filtros.

Mediana: En la figura 49 se observa la imagen luego de aplicar el filtro de la mediana. En OpenCV la función se la invoca de la siguiente manera:

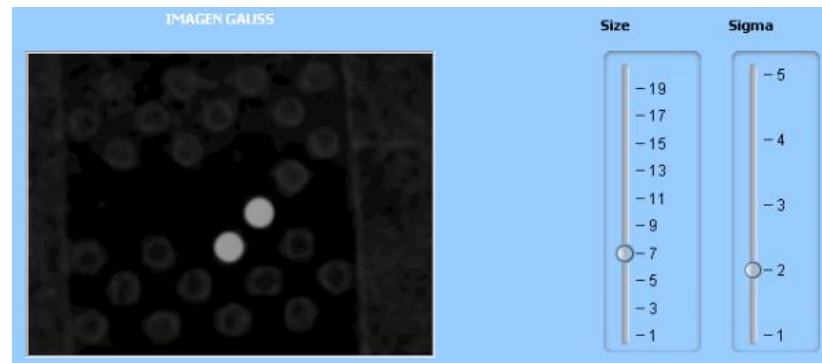
```
public void imageToMediana(Mat image,int size){
    setImageMediana(new Mat(image.width(), image.height(), image.type()));
    Imgproc.medianBlur(image, imageMediana, size);
}
```

Figura 48. Imagen luego de aplicar filtro de la mediana 15x15



Elaborado por: Edgar Changoluisa

Figura 49. Imagen luego de aplicar filtro gauss 7

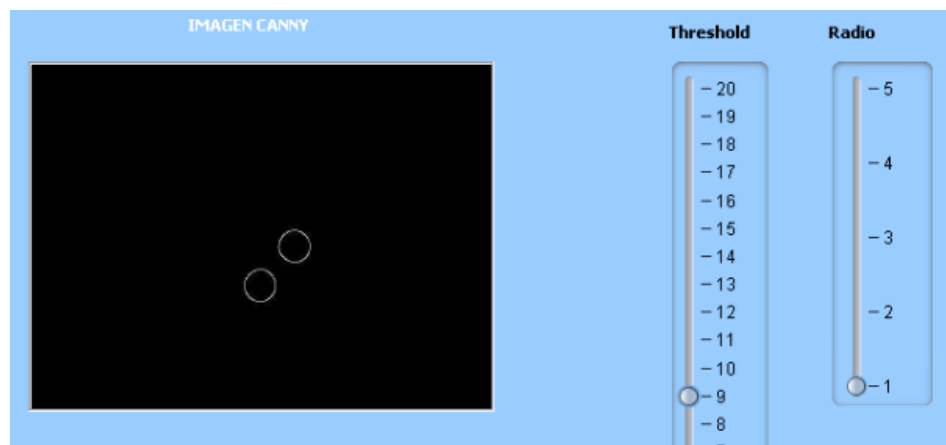


Elaborado por: Edgar Changoluisa

Gauss: En la figura 50 se observa la imagen luego de aplicar el filtro de Gauss. En OpenCV, la función se la invoca de la siguiente manera:

```
public void imageToImageGauss(Mat image,int size,double sigma) {  
    setImgGaus(new Mat(image.width(), image.height(), image.type()));  
  
    Size s = new Size(size, size);  
  
    sigma=sigma+0.5;  
  
    Imgproc.GaussianBlur(image, getImgGaus(), s, sigma, sigma);  
}
```

Figura 50. Obtención de bordes de la imagen



Elaborado por: Edgar Changoluisa

6.2.5. Detección de bordes.

Se realiza mediante la función `cvCanny` que como su nombre indica, calcula los contornos mediante el método de Canny. A esta función se le pasa una imagen en escala de grises o en formato binario y devuelve otra imagen con los bordes calculados. En la figura 51 se observa, la imagen resultado está en formato binario, toda la imagen es negra excepto los bordes que son blancos. El código para invocar a la función Canny de OpenCV es:

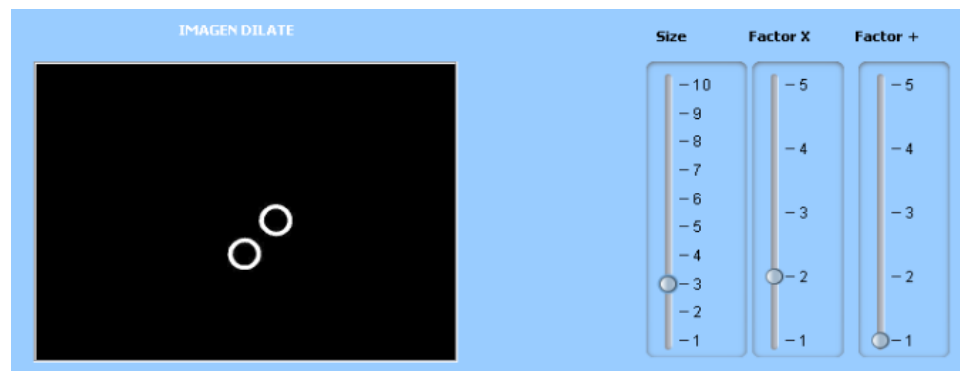
```
public void imageToCanny(Mat image, int min_threshold, int ratio) {  
    imgCanny = new Mat(image.width(), image.height(), image.type());  
    Imgproc.Canny(image, imgCanny, min_threshold, min_threshold * ratio);  
}
```

6.2.6. Dilatación de bordes.

Luego de dilatar los bordes de la imagen los objetos quedan bien diferenciados y queda la imagen preparada para pasar a la siguiente etapa. La función de opencv que dilata el borde de la imagen es:

```
public void imageToImageDilate(Mat image, int dilatacion_size, int factorMult, int  
factorSum) {  
    setImgDilate(new Mat(image.width(), image.height(), image.type()));  
    Size s = new Size(factorMult * dilatacion_size + factorSum, factorMult *  
dilatacion_size + factorSum);  
    Point p = new Point(dilatacion_size, dilatacion_size);  
    Mat element = Imgproc.getStructuringElement(Imgproc.MORPH_ELLIPSE, s, p);  
    Imgproc.dilate(image, getImgDilate(), element);  
}
```

Figura 51. Imagen con los bordes dilatados

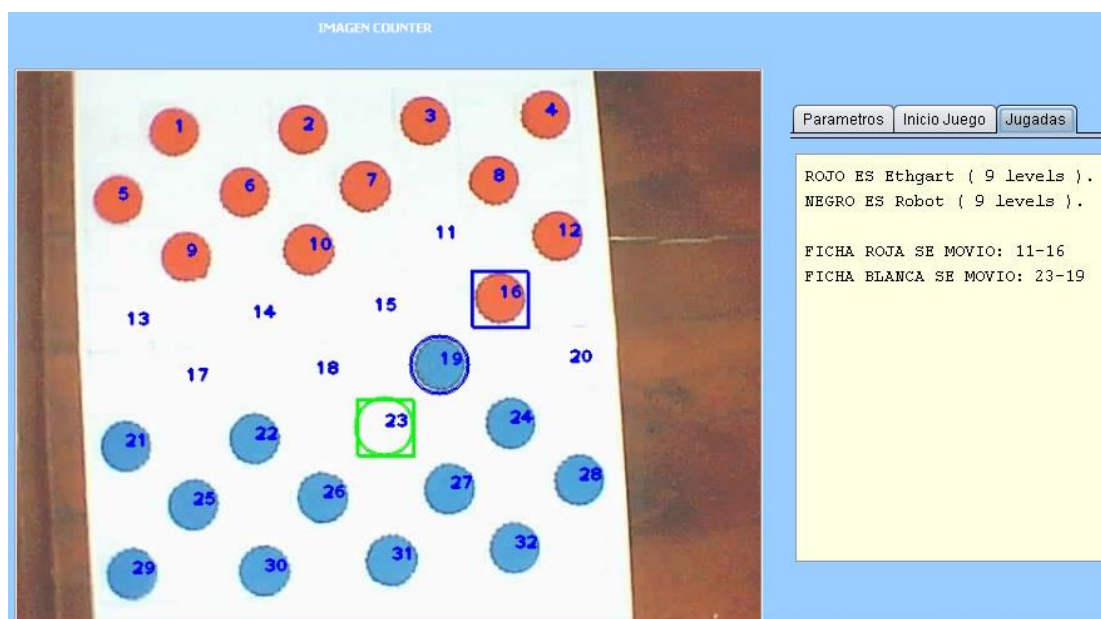


Elaborado por: Edgar Changoluisa

6.2.7. Reconocimiento de la jugada realizada.

Para determinar cuál fue la jugada que el contrincante humano realizo en el tablero de damas, se realizará una búsqueda de contornos, para lo cual la librería openCV tiene la función `cvContourMoments(contorno,& momentos)`, luego se procederá a obtener el valor de las coordenada en x,y para cada ficha que se haya movido.

Figura 52. Reconocimiento de una jugada realizada por el participante humano



Elaborado por: Edgar Changoluisa

A continuación se incluye el fragmento de código que se utiliza para reconocer la jugada que realizó el usuario:

```
public SearchFicha(Tablero tmptableroBase, Tablero tableroNuevo,
ArrayList<Set<String>> list,
    Mat tmpImage, boolean pisRed, boolean isNuevoJuego) {

    listPlantillas = list;
    image = tmpImage;
    CompararFichas compararFichas = new CompararFichas();
    moveFichas = new ArrayList<String>();

    for (int i = 1; i <= tmptableroBase.getNumFichas(); i++) {
        if (tmptableroBase.getFicha(i).getColor() == 3) {
            tmptableroBase.getFicha(i).setColor(0);
        }
        for (int j = 1; j <= tableroNuevo.getNumFichas(); j++) {
            if (tableroNuevo.getFicha(j).getColor() != 3 && (compararFichas.valirColicion(
                tmptableroBase.getFicha(i).getPoint(),
                tableroNuevo.getFicha(j).getPoint(),
                tmptableroBase.getFicha(i).getRadius(),
                tableroNuevo.getFicha(j).getRadius()) > 0)) {
                tmptableroBase.getFicha(i).setPoint(tableroNuevo.getFicha(j).getPoint());
                log.imprimir("cambio ficha : " + i + " color : " +
tmptableroBase.getFicha(i).getColor(), true);
                moveFichas.add("" + i + (tmptableroBase.getFicha(i).getColor() != 0 ? "" :
""));
                if (isNuevoJuego) {
                    if ((pisRed && tmptableroBase.getFicha(i).getColor() != 2)
                        || (!pisRed && tmptableroBase.getFicha(i).getColor() != 1)) {
                        tmptableroBase.getFicha(i).setCambio(true);
                    }

                    else if ((!pisRed && tmptableroBase.getFicha(i).getColor() == 1)
                        || (pisRed && tmptableroBase.getFicha(i).getColor() == 2)) {
                        tmptableroBase.getFicha(i).setColor(3);
                    }

                    if (tmptableroBase.getFicha(i).getColor() == 0) {
                        tmptableroBase.getFicha(i).setColor(pisRed ? 1 : 2);
                    }

                    else if (tmptableroBase.getFicha(i).getColor() != 0) {
                        tmptableroBase.getFicha(i).setColor(0);
                    }

                } else {
                    tmptableroBase.getFicha(i).setCambio(true);
                }
                break;
            }
        }
    }
}
```

6.2.8. Interfaz de usuario del sistema.

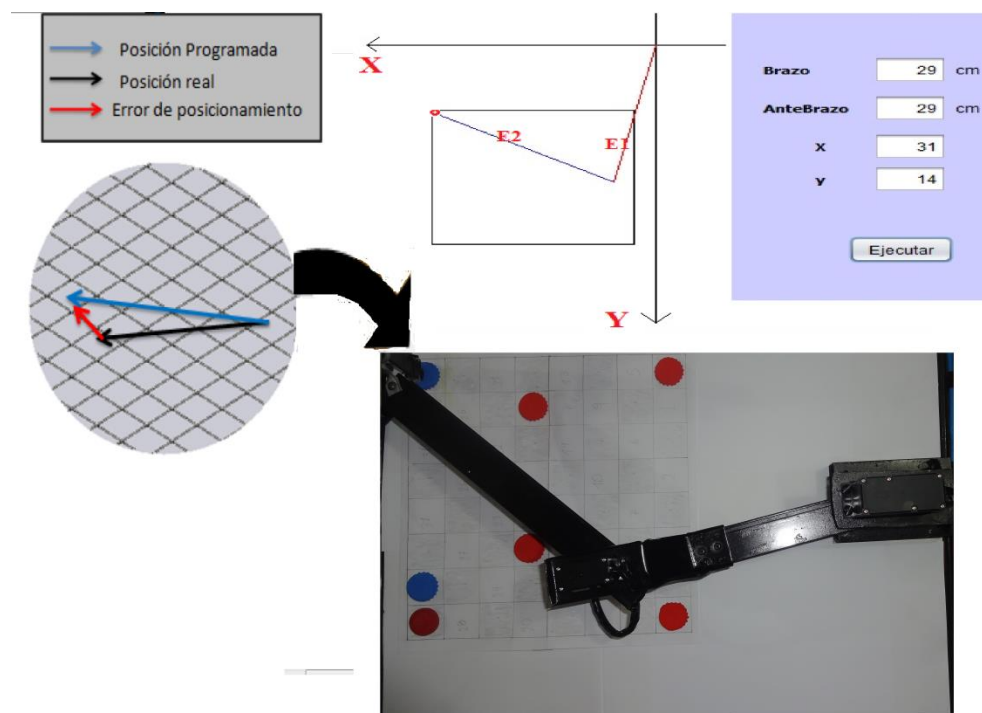
En la figura 53 se observa la pantalla que el usuario va a utilizar, a la izquierda se observa la imagen del tablero de damas capturada por la cámara web, en la que se indica cual fue el movimiento que realizo el usuario, a la derecha se observa un tab con la etiqueta parámetros en la cual el usuario puede configurar los parámetros del juego, en el tab de jugadas se registra el historial de los movimientos que han realizado el robot y el usuario.

6.3. Pruebas y ajustes

Las pruebas se va a dividir en dos partes: las pruebas de software y de hardware.

Pruebas de hardware

Figura 53. Pruebas de precisión de posicionamiento den el plano xy



Elaborado por: Edgar Changoluisa

Pruebas de precisión de posicionamiento en el plano xy del robot scara

Esta prueba realiza una comparación entre las coordenadas de posicionamiento programadas y las coordenadas reales, tomando como referencia una cuadrícula milimetrada que representa el plano xy en la cual se fija firmemente el robot,

procurando que el origen de coordenadas coincida con el eje de la primera articulación y se establecieron 5 puntos dentro del espacio de trabajo y de esta manera se determina el error de posicionamiento. En la figura 54 se observa la prueba realizadas con el prototipo.

Ecuación 13. *Cálculo de la precisión de posicionamiento*

$$App = \sqrt{((Xci - Xai)^2 + (Yci - Yai)^2)}$$

Fuente: (Martínez, 2012, pág. 111)

Tabla 19. *Pruebas de presión de posición del robot scara en el eje xy*

| Pruebas de precisión | | | | | |
|--|-----|------------------------|-------|----------------------------|------|
| coordenadas Programadas | | Coordenadas Alcanzadas | | Presión de posicionamiento | |
| Xci | Yci | Xai | Yai | | |
| 31 | 42 | 31,03 | 41,97 | 0,042426407 | |
| 31 | 14 | 31,05 | 14,01 | 0,050990195 | |
| 3 | 14 | 3 | 14,02 | 0,02 | |
| 3 | 42 | 3,05 | 41,08 | 0,921357694 | |
| 17 | 28 | 16,03 | 27,97 | 0,970463807 | |
| Presión de posicionamiento promedio (cm) | | | | | 0,40 |

Elaborado por: Edgar Changoluisa

En la tabla 19 se muestran los resultados de las prueba de presión que se realizaron con el prototipo de robot scara.

Durante las pruebas las coordenadas programadas, representan las posiciones a las que el prototipo debe movilizarse, y las coordenadas alcanzadas son aquellas en las que el prototipo realmente se posiciono, en un el caso ideal las coordenadas programadas son iguales a las alcanzadas, la diferencia entre las coordenadas programadas y las alcanzadas se obtiene con la ecuación 13. El valor promedio del campo presión de posicionamiento es de 0,40 cm, lo cual indica que el prototipo va a fallar en +- 0,4cm al posicionarse en una coordenada, el valor obtenido no se considera como error y se acepta tolerable para el proyecto.

Pruebas de software

Las pruebas funcionales del software del sistema se enfocan en los resultados finales emitidos por el sistema, estas pruebas analizan el desenvolvimiento del sistema vs el

entorno. Para validar los resultados de la prueba siempre se hace una comparación entre el resultado esperado vs el resultado obtenido.

A continuación se describen de manera detallada y explícita los resultados obtenidos de las pruebas realizadas al sistema en los siguientes casos:

Tabla 20. *Prueba del nivel óptimo del juego de damas*

| Nivel | Jugador Rojo | Max. Tiempo Jugada Robot | Total de jugadas | # Reina Roja | # Reina Azul | # Ficha Roja | # Ficha Azul | Evaluación |
|-------|--------------|--------------------------|------------------|--------------|--------------|--------------|--------------|------------|
| 1 | Robot | 5ms | 48 | 0 | 1 | 1 | 5 | -7 |
| 1 | Humano | 5ms | 55 | 2 | 0 | 3 | 1 | 8 |
| 2 | Robot | 10 ms | 54 | 1 | 0 | 1 | 0 | 4 |
| 2 | Humano | 10ms | 56 | 0 | 3 | 0 | 5 | -14 |
| 3 | Robot | 30ms | 54 | 0 | 1 | 0 | 5 | -8 |
| 3 | Humano | 30ms | 71 | 1 | 0 | 1 | 0 | 4 |
| 4 | Robot | 32ms | 58 | 0 | 2 | 1 | 3 | -8 |
| 4 | Humano | 40ms | 51 | 1 | 0 | 5 | 0 | 8 |
| 5 | Robot | 51ms | 54 | 1 | 1 | 3 | 0 | 3 |
| 5 | Humano | 50ms | 52 | 1 | 1 | 3 | 0 | 3 |
| 6 | Robot | 70ms | 50 | 1 | 1 | 0 | 0 | 0 |
| 6 | Humano | 40ms | 56 | 0 | 1 | 0 | 3 | -6 |
| 7 | Robot | 210ms | 51 | 1 | 2 | 0 | 1 | -4 |
| 7 | Humano | 160ms | 50 | 0 | 1 | 0 | 4 | -7 |
| 9 | Robot | 1071ms | 60 | 0 | 2 | 0 | 1 | -7 |
| 9 | Humano | 1181ms | 56 | 0 | 1 | 0 | 3 | -6 |
| 11 | Robot | 10826ms | 51 | 2 | 0 | 3 | 0 | 9 |
| 11 | Humano | 20404ms | 72 | 0 | 2 | 0 | 0 | -6 |

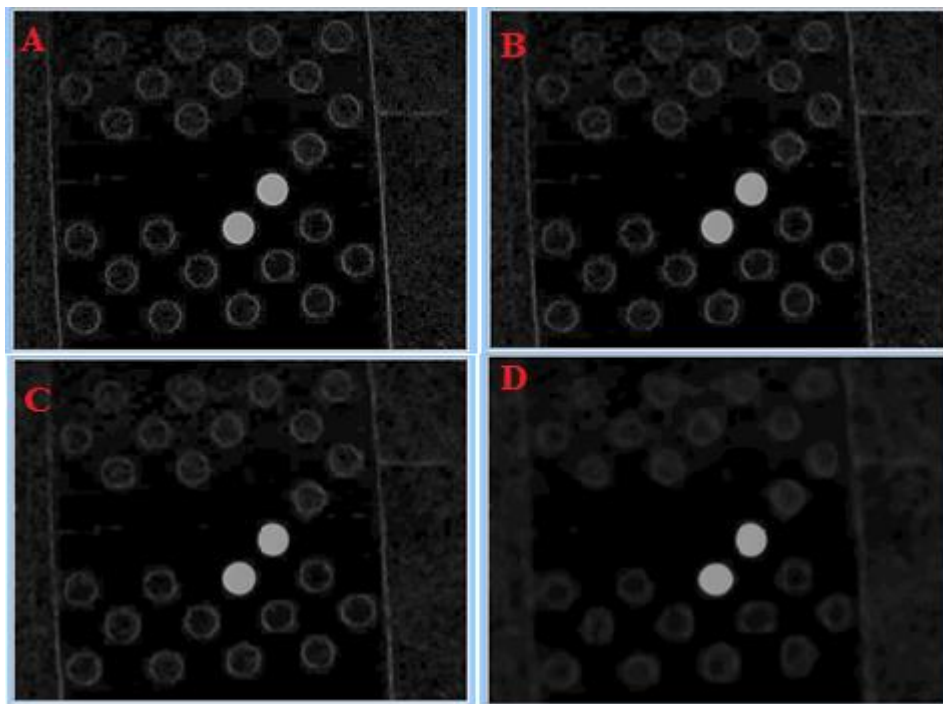
Elaborado por: Edgar Changoluisa

En la tabla 20 se describe el resultado de las pruebas realizadas con el algoritmo de búsqueda y selección del mejor movimiento en el juego de damas, a continuación se detalla la interpretación de cada columna de la tabla.

- **Nivel:** En el capítulo 3 en el ítem 3.4.1.3.2 de corte de la búsqueda, se explica que mientras mayor sea la profundidad del nivel de búsqueda, la jugada elegida va a hacer la mejor pero va a tomar más tiempo.
- **Jugador Rojo:** Este Jugador es el que realiza la primera jugada, en el anexo 1 que describe las reglas del juego de damas existe más detalles.

- **Max. Tiempo Jugada Robot:** Se indica que tiempo en milisegundos, que se demoró la búsqueda de la mejor jugada, esto va a depender del nivel de profundidad.
- **Total de jugadas:** Representa el número de movimientos que los dos jugadores realizaron durante la partida, hasta que alguien gane o se termine en tablas.
- **Función de evaluación:** Utilizando la ecuación 9 se evalúa qué nivel tiene más probabilidades de ganar, cuando el robot es el jugador que inicia la partida. El número de damas y fichas que tienen los dos jugadores, al final de la partida son los valores que se reemplazan en la mencionada ecuación.

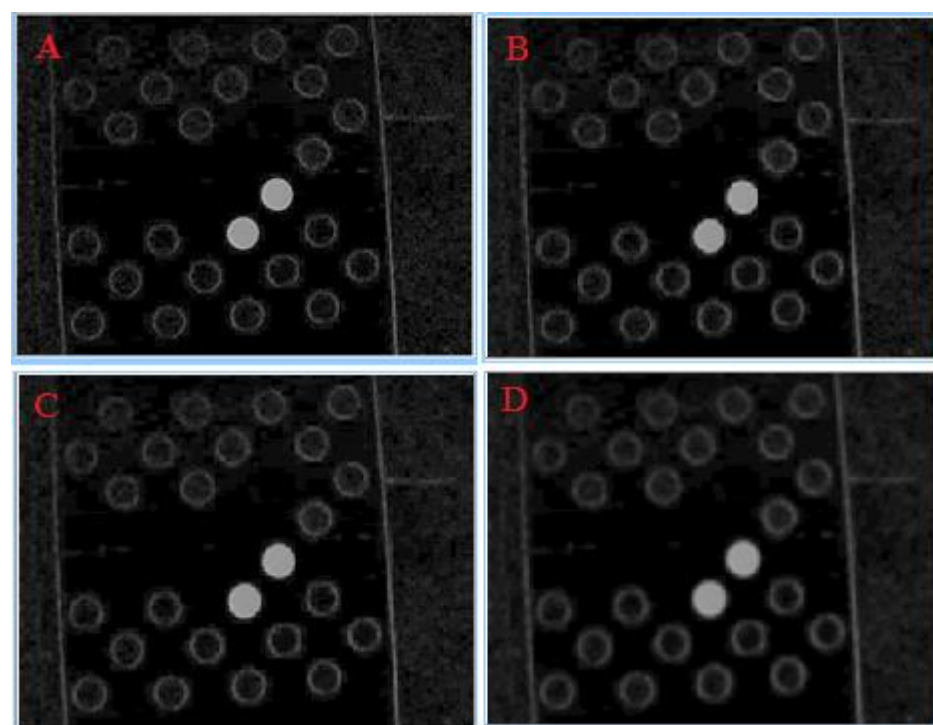
Figura 54. Pruebas del filtro de mediana



Elaborador por: Edgar Changoluisa.

En la figura 55 describe las pruebas realizadas de aplicación del filtro de la mediana en la imagen del tablero de damas, A. Matriz de 1x1, B. Matriz de 3x3, C. Matriz de 5x5, D. Matriz de 7x7. En las imágenes A, B y C existe gran cantidad de ruido. En el ítem 2.5.1.3 del capítulo 2 se encuentra más información acerca de este filtro.

Figura 55. Filtro de suavizado de Gauss

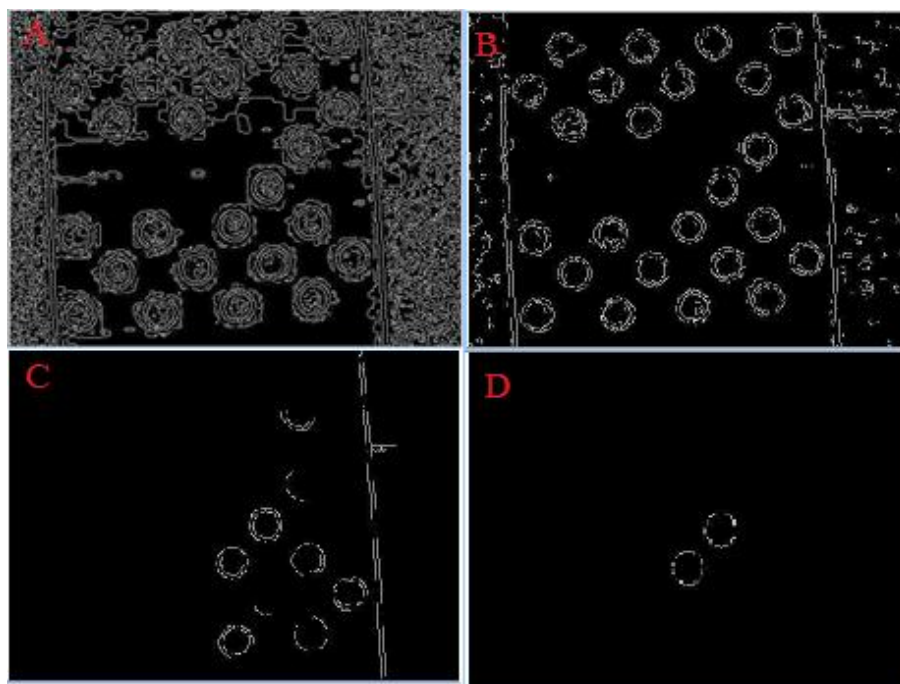


Elaborado por: Edgar Changoluisa.

En la figura 56 se presentan las imágenes de las pruebas realizadas luego de aplicar el filtro de gauss, los parámetros utilizados son $\sigma=2$: A. Matriz 1x1, B. Matriz 3x3, C. Matriz 5x5, D. Matriz 7x7. Se observa que la figura A, B no eliminan el ruido y D suaviza demasiado la imagen por lo cual se ve demasiado borrosa. En el ítem 2.5.1.4 del capítulo 2 se encuentra más información acerca de este filtro.

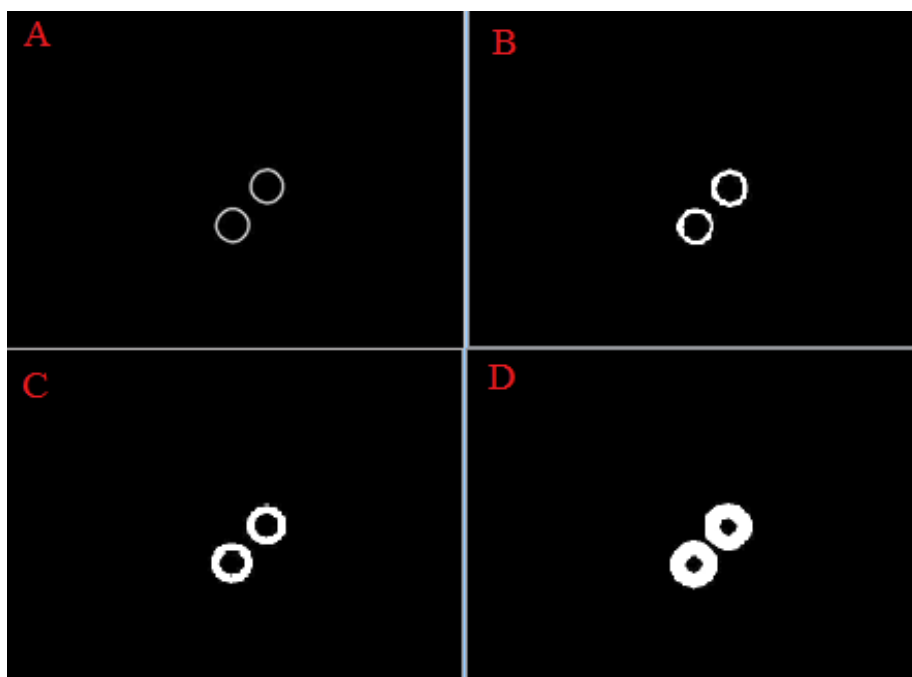
Para segmentar una imagen y etiquetarla el paso inicial, es buscar los bordes de las figuras, que contiene la imagen que se está procesando, en el proyecto se busca el borde de la fichas, para lo cual se utiliza el algoritmo de Canny con un radio 1 y modificando la intensidad de gradiente de la imagen: A. Gradiente 1, B. Gradiente 3, C. Gradiente 5, D. Gradiente 7. En las imágenes A, B, C se observa que con las configuraciones aplicadas, no se puede distinguir los bordes de las fichas, de los bordes de las otras figuras que son parte de la imagen que se está procesando.

Figura 56. Pruebas del algoritmo de Canny para buscar bordes



Elaborada por: Edgar Changoluisa

Figura 57. Pruebas de dilatación de bordes de las fichas



Elaborado por: Edgar Changoluisa.

Luego de obtener el borde de cada ficha, se observa en la figura 57.C, que el borde tiene líneas entrecortadas, por lo cual es necesario dilatar el borde para poder diferenciar cada fichas, obtener su ubicación y su radio, la figura 58 describe las pruebas realizadas, con diferentes parámetros como son: A.) matriz 1x1, B.) matriz 3x3, C.) matriz 5x5, D.) matriz 10x10. En las imágenes C y D las fichas se muestran con un radio mayor al real, esto puede presentar dificultad al momento de etiquetar las fichas, porque si las fichas se encuentran a una distancia un centímetro o menos entre sí, cuando se dilaten los bordes estos pueden intersecarse entre sí y dar la apariencia de que fueran una sola figura. En el capítulo 2 en el ítem 2.5.2.2 se describe más a detalle la dilatación del borde de una figura.

6.4. Resultados obtenidos

Luego de la realización de las pruebas se generaron nuevas configuraciones:

- El nivel de profundidad de búsqueda que se usa es 5, en el algoritmo que resuelve el juego de las damas, este valor se obtuvo luego del análisis de los parámetros de la tabla 20. Los parámetros evaluados fueron, el tiempo mínimo que el robot demora, en realizar cada jugada, el menor número de jugadas que realizó el robot para ganar la partida y la mayor probabilidad que el robot tiene de ganar el juego, cuando inicia la partida, este valor se lo obtiene con la función de evaluación.
- El filtro de medianas va a procesar la imagen con una matriz de 7x7, porque en las pruebas, se observó que esta configuración, elimina mejor el ruido de tipo sal y pimienta de la imagen procesada.
- El filtro de gauss va a procesar la imagen con una matriz de 5x5, porque si elegimos una matriz más grande, de acuerdo a las pruebas realizadas, se observa que la imagen resultante, es muy borrosa.
- El algoritmo de canny va a utilizar una intensidad del gradiente de la imagen de 7, porque con un valor menor, en las pruebas se observa que los bordes de las fichas no se diferencian de las demás figuras de la imagen.
- Una matriz de 3x3, es la mejor configuración para la dilatación del borde de cada ficha, para no perder el valor del radio original.

CONCLUSIONES

Mediante la realización de este trabajo de titulación se logró el cumplimiento del objetivo principal, el cual era diseñar y construir un prototipo del robot scara que pueda jugar damas por medio de visión artificial contra un ser humano.

Mediante la realización del trabajo se adquirió experiencia en la fase del diseño y construcción, de los componentes mecánicos de un robot scara, resaltando detalles importantes como: buscar el material más liviano y resistente para los eslabones, por lo cual se eligió aluminio. Se utiliza servomotores para mover cada eslabón porque tiene precisión, torque y bajo consumo de energía.

Se construyó y diseño un prototipo de brazo robótico scara de bajo costo, con un valor de 250,40 dólares americanos, la descripción de los materiales y el valor de cada uno de ellos, se describen en el anexo 3. El valor mínimo para un artículo de este tipo es 707,20, según la página web ebay, que es el mayor centro de compra y venta en Internet.

Se logró reconocer las jugadas realizadas en el tablero de damas, por el contrincante humano, en un ambiente controlado, para lo cual se utilizó un sistema de iluminación y además una tela de color negra para bloquear el exceso de luz del exterior.

Con la finalidad de obtener las mejores jugadas que se deben realizar durante el juego, se diseñó e implantó un sistema inteligente, el cual utiliza un algoritmo de análisis y búsqueda en tiempo real.

El tiempo de respuesta del sistema es menor a 1 segundo, para el nivel 5 de profundidad en la búsqueda del algoritmo de la poda alfa & beta, este valor se va a usar, como default en este proyecto, porque luego de las pruebas realizadas con el algoritmo de búsqueda, mismas que se describen en la tabla 20, se concluye que el nivel 5 presenta un buen nivel de complejidad para el adversario.

LISTA DE REFERENCIAS

- Abásolo, M. J. (2001). *Introducción a la Inteligencia Artificial*. Recuperado el 31 de enero de 2014, de <http://dmi.uib.es/~abasolo/intart/2-juegos.html#2.1>.
- Aliane, N. (2004). Un robot que juega a las damas. *IEEE*, 5.
- Aritor. (2008). *Teorema del coseno*. Obtenido de http://www.aritor.com/trigonometria/teorema_coseno.html
- Boada, Y. F., & Morales, L. A. (mayo de 2010). *Construcción, ensamblaje y control de un robot de tres grados de libertad tipo scara*. Recuperado el 28 de enero de 2014, de <http://bibdigital.epn.edu.ec/bitstream/15000/2219/1/CD-2976.pdf>
- Brennan, J. (2014). *Diferencias entre las reflexiones especulares y difusas*. Recuperado el 01 de febrero de 2014, de http://www.ehowenespanol.com/diferencias-reflexiones-especulares-difusas-info_268711/
- Checa, D., Luna, D., & Mosquera, V. (2009). *Simulador de un Robot SCARA de 4 grados de Libertad basado en realidad virtual*. Recuperado el 28 de enero de 2014, de http://www.academia.edu/3811178/Simulador_de_un_Robot_SCARA_de_4_Grados_de_Libertad_Basado_en_Realidad_Virtual
- Colegio Teodoro Alvarado Olea. (2010). *Definición: Inteligencia artificial*. Recuperado el 31 de enero de 2014, de <http://colegiotao.webnode.es/inteligencia-artificial/>
- Control Systems Principles. (2009). *Sistemas de servo control 2: Servomecanismos digitales*. Recuperado el 4 de enero de 2014, de <http://www.control-systems-principles.co.uk/whitepapers/spanishwp/02ServoPaper2SP.pdf>
- Esqueda, J. J., & Palafox, L. E. (2005). *Fundamentos de procesamiento de imágenes*. Mexicali: Universidad Autónoma de Baja California. III.t.

- Fuente, E., & Trespaderne, F. M. (2012). Visión Artificial industrial. En E. Fuente López, & F. M. Trespaderne. Valladolid: Universidad de Valladolid.
- Guzmán, A. E., & Sevillano, E. S. (20 de agosto de 2012). *Diseño y construcción de un robot scara de tres grados de libertad con un efector final electromagnético, para el paletizado de recipientes metálicos*. Recuperado el 12 de febrero de 2014, de <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CCYQFjAA&url=http%3A%2F%2Fmecatronica.espe.edu.ec%2Fwp-content%2Fuploads%2F2012%2F10%2F%25E2%2580%259CDISE%25C3%2591O-Y-CONSTRUCCI%25C3%2593N-DE-UN-ROBOT-SCARA-DE-TRES-GRA-DOS-DE>
- Hernández, E. (2006). Cómo escribir una tesis. Escuela Nacional de salud pública.
- IDLServicios.com. (2014). *Informatica moderna*. Recuperado el 31 de enero de 2014, de http://www.informaticamoderna.com/Camara_web.htm
- IEEE. (11 de 2008). Una experiencia práctica de aprendizaje basado en proyecto en una asignatura de robótica. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 95.
- Iñigo, R., & Vidal, E. (2002). *Robots industriales manipuladores*. Catalunya: Universidad Politécnica de Catalunya.
- Martínez, D. A. (2012). *Diseño y construccion de prototipos de robot scara 3 DOF*. Recuperado el 28 de enero de 2014, de <https://repositorio.uis.edu.co/jspui/bitstream/123456789/6100/2/145391.pdf>
- Ollero, A. (2001). *Robótica: manipuladores y robots móviles*. Barcelona: MARCOMBO S.A.
- Opencv dev team. (31 de diciembre de 2013). *Canny Edge Detector: OpenCV 2.4.8.0 documentation*. Recuperado el 31 de enero de 2014, de

http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html#canny-detector

Proyectos Robóticos. (2008). *Proyectosroboticos:cinematica-inversa-i*. Recuperado el 28 de enero de 2014, de

<https://sites.google.com/site/proyectosroboticos/cinematica-inversa-i>

robotshop. (2014). *Robot Arm Torque Calculator*. Recuperado el 30 de enero de 2014, de <http://www.robotshop.com/blog/en/robot-arm-torque-calculator-9712>

Russell, S. J., & Norvig, P. (2004). *Inteligencia Artificial un enfoque moderno*. En S. J. Norvig. Madrid: Pearson Educación S.A.

Sobrado, E. A. (2003). *Sistema de Visión Artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot*. Lima, Peru.

Sotomayor , N. (septiembre de 2009). *Departamento de Automatización y Control Industrial*. Recuperado el 24 de enero de 2014, de <http://ciecfie.epn.edu.ec/Material/4toNivel/Robotica/ROBOTICA%20INDUSTRIAL.pdf>

Sparx Systems. (2007). *Diagrama de Despliegue UML 2*. Recuperado el 15 de enero de 2014, de http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html

TecMikro. (01 de febrero de 2014). *TecMikro:El servomotor*. Recuperado el 03 de enero de 2014, de <http://www.programarpicenc.com/productos/servomotor.html>

Universidad de Sevilla. (2013). *Dep. Matemática Aplicada: Procesamiento de imágenes digitales*. Recuperado el 31 de enero de 2014, de <http://alojamientos.us.es/gtocom/pid/tema3-1.pdf>

Universidad Nacional de Quilmes. (septiembre de 2005). *Filtrado Espacial*.

Obtenido de

<http://iaci.unq.edu.ar/materias/vision/archivos/apuntes/Filtrado%20Espacial.pdf>

Universidad Nacional de Quilmes. (agosto de 2005). *Iluminación para las*

aplicaciones de Visión Artificial. Obtenido de

http://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CDIQFjAB&url=http%3A%2F%2Fiaci.unq.edu.ar%2Fmaterias%2Fvision%2Farchivos%2Fapuntes%2FTipos%2520de%2520Iluminaci%25C3%25B3n.pdf&ei=QE8WU6epCqWF0QGR3ICgAg&usg=AFQjCNFHlAgiiWBcTu-T_0N-2R1

Universidad Nacional de Quilmes. (octubre de 2005). *Ing. en Automatización y*

Control Industrial. Recuperado el 29 de enero de 2014, de

<http://iaci.unq.edu.ar/materias/vision/archivos/apuntes/Aspectos%20de%20un%20Proyecto%20de%20Visi%C3%B3n%20Artificial.pdf>

Universidad Nacional de Quilmes. (octubre de 2005). *Operaciones Morfológicas en*

Imágenes Binarias. Obtenido de

<http://iaci.unq.edu.ar/materias/vision/archivos/apuntes/Operaciones%20Morfol%C3%B3gicas%20en%20Im%C3%A1genes%20Binarias%20-%20parte%201.pdf>

Vélez, J. F., Moreno, A. B., Sánchez, Á., & Sánchez, J. L. (2007). *Visión por computador* (2ª ed.). Universidad Rey.

visionartificial.fpcat.cat. (marzo de 2012). *Conocimientos básicos de visión*

artificial: Visión Artificial. Recuperado el 31 de enero de 2014, de

<http://visionartificial.fpcat.cat/wp-content/uploads/Conocimientos.pdf>

Anexo 1: Reglas de las damas

El juego de las damas consta de 24 peones divididos en 12 blancos y 12 negros y un tablero de 64 casillas (8x8) coloreadas alternamente blancas y negras.

Objetivo

La finalidad del juego es la captura o bloqueo de todas las piezas contrarias, de forma que no les sea posible realizar movimiento.

Desarrollo del juego

Cada jugador controla las piezas de un color situadas al comienzo a cada lado del tablero. Empieza el juego las blancas,

Los movimientos se hacen alternativamente, uno por jugador, en diagonal, una sola casilla y en sentido de avance, o sea, hacia el campo del oponente.

Si un jugador consigue llevar una de sus fichas al lado contrario del tablero cambiará dicho peón por una dama o reina.

La dama o reina se mueve también en diagonal, pero puede hacerlo hacia delante y hacia atrás. Según las opciones de mesa puede avanzar una casilla como el peón o recorrer cualquier número de casillas mientras estén libres. Nunca podrá saltar por encima de sus propias piezas o dos piezas contiguas.

Capturar fichas del contrario

Se pueden capturar una ficha del contrario, si esta se encuentra delante de una de tus piezas y es posible saltar por encima de ella siempre en dirección de ataque y en diagonal, y caer en la casilla inmediatamente detrás de aquella.

La captura con dama o reina en una de sus variantes es igual que con el peón con la diferencia que puede capturar en cualquier dirección. Según se especifique en las opciones de mesa la ficha chaparrada no tiene porque estar en la casilla contigua

mientras esté en la misma diagonal, sin otras fichas entre la dama que realiza la captura y la ficha capturada.

La captura es obligatoria. Si una o más piezas están en situación de realizar capturas, será obligatorio realizar tal captura, no pudiendo optar por mover otra pieza.

Una vez realizada una captura, tanto un peón como una dama deben volver a capturar si es posible según las reglas anteriores y en el mismo turno del jugador, y así sucesivamente.

Ley de la cantidad: es obligatorio capturar al mayor número posible de fichas.

Ley de la calidad: a igual número de fichas capturar, es obligatorio capturar a las de mayor calidad, dama antes que peón.

Final de partida:

Se llega al final de partida cuando un jugador abandona, se queda sin fichas o estas no tienen posibilidad de movimiento (bloqueo o ahogada).

Se llega a tablas o empate cuando ambos jugadores lo pactan, cuando se lleven las jugadas indicadas por el master sin desplazamiento de peón ni captura.

Anexo 2: Lista de materiales para el circuito de la tarjeta madre

| Listas materia de la Fuente de Alimentación | cantidad |
|---|----------|
| condensador electrolítico 4700 uf | 2 |
| condensador cerámico 10nf | 2 |
| LM2576ADJ | 2 |
| Diodo BYV28 | 2 |
| Resistencia 82114R | 2 |
| Condensador electrolítico 1000UF | 2 |
| Resistencia 1K | 2 |
| Potenciómetro 5K/10K | 2 |
| LM7805 | 1 |
| Listas materia de la tarjeta de Control | cantidad |
| Micro controlador PIC16F877 | 1 |
| MAX232 | 1 |
| Condensador Cerámico 1UF | 4 |
| Resistencia 1K | 5 |

Elaborado por: Edgar Changoluisa.

Anexo 3: Costo del diseño y construcción del prototipo del brazo robótico scara.

| ELEMENTOS Y MATERIALES | VALOR |
|---|---------------|
| Hitec 31805S HS-805BB Mega Giant Scale 2BB Servo | 47,22 |
| Hitec 31081S HS-81 Sub Micro Servo | 15,65 |
| Hitec 35485S HS-5485HB Standard Servo | 25,03 |
| 4 Rodamientos | 4,00 |
| 60 cm Tubo cuadrado de hierro | 2,30 |
| Materiales electrónicos | 16,20 |
| Diseño y construcción circuito electrónico de control | 70,00 |
| Diseño y construcción de la estructura mecánica del robot | 50,00 |
| Programación del microcontrolador | 20,00 |
| TOTAL | 250,40 |

Elaborado por: Edgar Changoluisa.